SQL Anywhere - Ultra Light
文書バージョン: 17 – 2016-05-11

Ultra Light Java API リファレンス

目次

1 Ultra LightJ API リファレンス

Ultra LightJ には豊富な API オブジェクトがあります。

次に、よく使用される API オブジェクトの一部を示します。

DatabaseManager

データベースと接続を管理する方法を提供します。

接続

Ultra Light データベースへの接続を表します。Connection オブジェクトは 1 つまたは複数作成できます。

SyncParms

Ultra Light データベースを Mobile Link サーバと同期させます。

PreparedStatement、ResultSet

動的 SQL 文の作成、クエリの記述、INSERT、UPDATE、DELETE 文の実行、プログラムによるデータベースの結果セットの制御を行います。

パッケージ [Android]

```
com.sap.ultralitejni17
```

> **注記**
>
> 主な SQL Anywhere マニュアルをお探しですか。マニュアルをローカルにインストールした場合は、Windows のスタートメニューを使用してアクセスするか (Microsoft Windows)、`C:¥Program Files¥SQL Anywhere 17¥Documentation` にナビゲートします。
>
> また、DocCommentXchange の Web で、主な SQL Anywhere API リファレンスマニュアルにアクセスすることもできます。http://dcx.sap.com

このセクションの内容:

Configuration インタフェース [28 ページ]
データベース用の Configuration オブジェクトを確立します。

Connection インタフェース [31 ページ]
データベース接続を表します。データベース操作を開始するには接続が必要です。

DatabaseInfo インタフェース [58 ページ]
Connection オブジェクトに関連付けられ、データベース情報を公開するメソッドを提供します。

DatabaseManager クラス [62 ページ]
基本設定を取得したり、新しいデータベースを作成したり、既存のデータベースに接続したりするための静的メソッドを提供します。

DecimalNumber インタフェース [69 ページ]
正確な decimal 値を表し、java.math.BigDecimal を使用できない Java プラットフォームに 10 進法計算のサポートを提供します。

Domain インタフェース [74 ページ]
テーブル内のカラムの Domain オブジェクトの型情報を表します。

FileTransfer インタフェース [77 ページ]
クライアントと Mobile Link サーバ間のファイル転送メカニズムを提供します。

FileTransferProgressData インタフェース [101 ページ]
ファイル転送の進行状況のモニタリングデータを通知します。

FileTransferProgressListener インタフェース [103 ページ]
ファイル転送の進行状況イベントを受信します。

IndexSchema インタフェース [105 ページ]
インデックスのスキーマを指定し、システムテーブルの問い合わせに便利な定数を提供します。

PreparedStatement インタフェース [106 ページ]
SQL クエリを実行して ResultSet オブジェクトを生成するか、準備された SQL 文をデータベースに対して実行するメソッドを提供します。

ResultSet インタフェース [130 ページ]
テーブルをローごとにトラバースし、カラムデータにアクセスするメソッドを提供します。

ResultSetMetadata インタフェース [160 ページ]
ResultSet オブジェクトに関連付けられ、カラム情報を提供するメソッドが含まれます。

SQLInfo インタフェース [168 ページ]
実行された SQL 文についての情報を示します。

StreamHTTPParms インタフェース [171 ページ]
HTTP を使用して Mobile Link サーバと通信する方法を定義する HTTP ストリームパラメータを表します。

StreamHTTPSParms インタフェース [184 ページ]
セキュア HTTPS 接続を使用して Mobile Link サーバと通信する方法を定義する HTTPS ストリームパラメータを表します。

ストリーム TCPIPParms インタフェース [191 ページ]
TCP/IP を使用して Mobile Link サーバと通信する方法を定義する TCP/IP ストリームパラメータを表します。

StreamTLSParms インタフェース [192 ページ]
セキュア TLS 接続を使用して Mobile Link サーバと通信する方法を定義する TLS ストリームパラメータを表します。

SyncObserver インタフェース [194 ページ]
同期の進行状況の情報を受け取ります。

SyncObserver.States インタフェース [196 ページ]
observer に通知できる同期ステータスを定義します。

SyncParms クラス [198 ページ]
データベース同期処理中に使用されたパラメータを保持します。

SyncResult クラス [224 ページ]
指定されたデータベース同期のステータス関連の情報をレポートします。

SyncResult.AuthStatusCode インタフェース [240 ページ]
Mobile Link サーバから返された認証コードを列挙します。

TableSchema インタフェース [241 ページ]
テーブルのスキーマを指定し、システムテーブルの名前を定義する定数を提供します。

ULjEvent インタフェース [242 ページ]
Ultra Light J API システムイベントを示します。

ULjException クラス [244 ページ]
データベースからスローされた例外に取って代わります。

Unsigned64 クラス [247 ページ]
符号なし 64 ビットのバイナリ値を実装します。

UUIDValue インタフェース [254 ページ]
ユニーク識別子 (UUID またはユニバーサルユニーク識別子) オブジェクトを記述します。

ValidateDatabaseProgressData インタフェース [257 ページ]
ValidateDatabase プログレスデータをレポートします。

ValidateDatabaseProgressData.StatusId インタフェース [258 ページ]
Ultra Light データベース検証ユーティリティのステータス ID を指定します。

ValidateDatabaseProgressListener インタフェース [260 ページ]
ValidateDatabase プログレスイベントを受信します。

1.1 ColumnSchema インタフェース

カラムのスキーマを指定します。

構文

```
public interface ColumnSchema
```

メンバー

ColumnSchema のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

変数

変更子とタイプ	変数	説明
public static final byte	COLUMN_DEFAULT_NONE	カラムにデフォルト属性が存在しない事を示します。 デフォルト属性を指定しない場合、次のデフォルト値が適用されます。 • NULL 入力可のカラムのデフォルト値は NULL • NULL 入力不可の数値カラムのデフォルト値は 0 • NULL 入力不可の可変長カラムのデフォルト値は長さ 0 の値 既存のテーブルのデフォルト値は、システムテーブル TableSchema.SYS_COLUMNS の column_default カラムに問い合わせることで確認できます。
public static final byte	COLUMN_DEFAULT_AUTOINC	AUTOINCREMENT カラムのデフォルト属性を示します。 AUTOINCREMENT 属性を使用する場合、カラムは整数データ型の 1 つ、または真数値型にします。INSERT 時に AUTOINCREMENT のカラムに値を指定しないと、カラム内のほかの値より大きいユニーク値が生成されます。INSERT で、カラムの現在の最大値より大きい値を指定した場合、この値が後続の挿入処理の開始ポイントとして使用されます。 Ultra Light J では、テーブルが作成された時点でのオートインクリメントの初期値は 0 ではありません。カラムに符号付きデータ型が指定されている場合は、AUTOINCREMENT 属性によって負の値が生成されます。このため、オートインクリメントを適用するカラムを符号なし整数として宣言し、負の値が生成されないようにしてください。 既存のテーブルのデフォルト値は、システムテーブル TableSchema.SYS_COLUMNS の column_default カラムに問い合わせることで確認できます。

変更子とタイプ	変数	説明
public static final byte	COLUMN_DEFAULT_GLOBAL_AUTOINC	GLOBAL AUTOINCREMENT カラムのデフォルト属性を示します。 この定数は AUTOINCREMENT 属性と同じですが、ドメインはパーティションに分割されます。各分割には同じ数の値が含まれます。データベースの各コピーにユニークなグローバルデータベース ID 番号を割り当てる必要があります。Ultra Light J では、データベースのデフォルト値は、そのデータベース番号でユニークに識別された分割から設定されます。 既存のテーブルのデフォルト値は、システムテーブル TableSchema.SYS_COLUMNS の column_default カラムに問い合わせることで確認できます。
public static final byte	COLUMN_DEFAULT_CURRENT_DATE	カラムのデフォルト属性が current date (年、月、日) である事を示します。 SQL Anywhere のマニュアルセットで、"Ultra Light の特別値" の下の "CURRENT DATE 特別値" を参照してください。 既存のテーブルのデフォルト値は、システムテーブル TableSchema.SYS_COLUMNS の column_default カラムに問い合わせることで確認できます。
public static final byte	COLUMN_DEFAULT_CURRENT_TIME	CURRENT TIME カラムのデフォルト属性を示します。 SQL Anywhere のマニュアルセットで、"Ultra Light の特別値" の下の "CURRENT TIME 特別値" を参照してください。 既存のテーブルのデフォルト値は、システムテーブル TableSchema.SYS_COLUMNS の column_default カラムに問い合わせることで確認できます。

変更子とタイプ	変数	説明
public static final byte	COLUMN_DEFAULT_CURRENT_TIMESTAMP	CURRENT TIMESTAMP カラムのデフォルト属性を示します。 この定数は、CURRENT DATE と CURRENT TIME の値を結合して、TIMESTAMP 値を形成します。この値は、年、月、日、時、分、秒、秒の小数位で構成されます。秒の精度は小数点以下第 3 位に設定されます。この定数の精度はシステムクロックの精度によって制限されます。 SQL Anywhere のマニュアルセットで、"Ultra Light の特別値" の下の "CURRENT TIMESTAMP 特別値" を参照してください。 既存のテーブルのデフォルト値は、システムテーブル TableSchema.SYS_COLUMNS の column_default カラムに問い合わせることで確認できます。
public static final byte	COLUMN_DEFAULT_UNIQUE_ID	カラムのデフォルト属性が新しいユニーク ID である事を示します。 UUID を使用して、テーブルのローをユニークに識別できます。生成される値は、すべてのコンピュータまたはデバイスでユニークになります。つまり、同期やレプリケーション環境でキーとして使用できます。 既存のテーブルのデフォルト値は、システムテーブル TableSchema.SYS_COLUMNS の column_default カラムに問い合わせることで確認できます。
public static final byte	COLUMN_DEFAULT_CONSTANT	カラムのデフォルト属性が constant である事を示します。 既存のテーブルのデフォルト値は、システムテーブル TableSchema.SYS_COLUMNS の column_default カラムに問い合わせることで確認できます。

変更子とタイプ	変数	説明
public static final byte	COLUMN_DEFAULT_CURRENT_UTC_TIMESTAMP	CURRENT UTC TIMESTAMP カラムのデフォルト属性を示します。 この定数は、CURRENT DATE と CURRENT TIME の値を結合して、UTC TIMESTAMP 値を形成します。この値は、GMT での年、月、日、時、分、秒、秒の小数位で構成されます。秒の精度は小数点以下第 3 位に設定されます。この定数の精度はシステムクロックの精度によって制限されます。 SQL Anywhere のマニュアルセットで、"Ultra Light の特別値" の下の "CURRENT UTC TIMESTAMP 特別値" を参照してください。 既存のテーブルのデフォルト値は、システムテーブル TableSchema.SYS_COLUMNS の column_default カラムに問い合わせることで確認できます。
public static final byte	COLUMN_DEFAULT_AUTOFILENAME	AUTOFILENAME カラムのデフォルト属性を示します。 VARCHAR カラムにこのデフォルト値がある場合、カラムは外部 blob 定義のファイル名のカラムになります。 カラムにこのタイプのデフォルトがある場合、システムテーブル TableSchema.SYS_COLUMNS の column_default_value カラムには、外部 blob 定義にあるプレフィクスと拡張子の文字列が 'prefix|extension' の形式で含まれます。 既存のテーブルのデフォルト値は、システムテーブル TableSchema.SYS_COLUMNS の column_default カラムに問い合わせることで確認できます。

備考

このインタフェースには、システムテーブル syscolumn の column_default カラムに格納されるさまざまなカラムデフォルト値の定数のみが含まれています。

1.2 ConfigFile インタフェース

ファイルに保存される永続的なデータベース用の Configuration オブジェクトを確立します。

構文

```
public interface ConfigFile extends ConfigPersistent
```

メンバー

ConfigFile のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

ConfigPersistent から継承されたメンバー

変更子とタイプ	メンバー	説明
public void	enableAesDBEncryption() [22 ページ]	データベースの AES 暗号化を可能にします。
public void	enableObfuscation() [22 ページ]	データベースの難読化を有効にします。
public int	getCacheSize() [22 ページ]	データベースのキャッシュサイズ (バイト単位) を返します。
public String	getConnectionString() [23 ページ]	setConnectionString メソッドで登録された接続文字列を取得します。
public String	getCreationString() [23 ページ]	setCreationString メソッドで登録された作成文字列を取得します。
public String	getEncryptionKey() [24 ページ]	setEncryptionKey メソッドで登録されたデータベース暗号化キーを取得します。
public String	getUserName() [24 ページ]	setUserName メソッドで設定されたユーザの名前を取得します。
public ConfigPersistent	setCacheSize(int) [25 ページ]	データベースのキャッシュサイズ (バイト単位) を設定します。
public void	setConnectionString(String) [26 ページ]	データベースの作成または接続に使用される接続文字列を設定します。
public void	setCreationString(String) [26 ページ]	データベースを作成するために使用される作成文字列を設定します。
public void	setEncryptionKey(String) [27 ページ]	暗号化キーを設定します。
public void	setUserName(String) [27 ページ]	ユーザの名前を設定します。

Configuration から継承されたメンバー

変更子とタイプ	メンバー	説明
public String	getDatabaseName() [29 ページ]	データベース名を返します。

変更子とタイプ	メンバー	説明
public int	getPageSize() [29 ページ]	データベースのページサイズ (バイト単位) を返します。
public Configuration	setDatabaseName(String) [29 ページ]	データベース名を設定します。
public Configuration	setPageSize(int) [30 ページ]	データベースのページサイズを設定します。
public Configuration	setPassword(String) [30 ページ]	データベースのパスワードを設定します。

関連情報

DatabaseManager クラス [62 ページ]

1.3 ConfigFileAndroid インタフェース

ファイルに保存される永続的なデータベース用の Configuration オブジェクトを確立します。

構文

```
public interface ConfigFileAndroid extends ConfigFile
```

メンバー

ConfigFileAndroid のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

ConfigPersistent から継承されたメンバー

変更子とタイプ	メンバー	説明
public void	enableAesDBEncryption() [22 ページ]	データベースの AES 暗号化を可能にします。
public void	enableObfuscation() [22 ページ]	データベースの難読化を有効にします。
public int	getCacheSize() [22 ページ]	データベースのキャッシュサイズ (バイト単位) を返します。
public String	getConnectionString() [23 ページ]	SetConnectionString メソッドで登録された接続文字列を取得します。
public String	getCreationString() [23 ページ]	SetCreationString メソッドで登録された作成文字列を取得します。

変更子とタイプ	メンバー	説明
public String	getEncryptionKey() [24 ページ]	setEncryptionKey メソッドで登録されたデータベース暗号化キーを取得します。
public String	getUserName() [24 ページ]	setUserName メソッドで設定されたユーザの名前を取得します。
public ConfigPersistent	setCacheSize(int) [25 ページ]	データベースのキャッシュサイズ (バイト単位) を設定します。
public void	setConnectionString(String) [26 ページ]	データベースの作成または接続に使用される接続文字列を設定します。
public void	setCreationString(String) [26 ページ]	データベースを作成するために使用される作成文字列を設定します。
public void	setEncryptionKey(String) [27 ページ]	暗号化キーを設定します。
public void	setUserName(String) [27 ページ]	ユーザの名前を設定します。

Configuration から継承されたメンバー

変更子とタイプ	メンバー	説明
public String	getDatabaseName() [29 ページ]	データベース名を返します。
public int	getPageSize() [29 ページ]	データベースのページサイズ (バイト単位) を返します。
public Configuration	setDatabaseName(String) [29 ページ]	データベース名を設定します。
public Configuration	setPageSize(int) [30 ページ]	データベースのページサイズを設定します。
public Configuration	setPassword(String) [30 ページ]	データベースのパスワードを設定します。

備考

ConfigFileAndroid インタフェースを実装するオブジェクトは、DatabaseManager.createConfigurationFileAndroid メソッドを使用して作成されます。

関連情報

DatabaseManager クラス [62 ページ]
createConfigurationFileAndroid(String, android.content.Context) メソッド [65 ページ]

1.4 ConfigPersistent インタフェース

永続的なデータベース用の Configuration オブジェクトを確立します。

構文

```
public interface ConfigPersistent extends Configuration
```

メンバー

ConfigPersistent のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public void	enableAesDBEncryption() [22 ページ]	データベースの AES 暗号化を可能にします。
public void	enableObfuscation() [22 ページ]	データベースの難読化を有効にします。
public int	getCacheSize() [22 ページ]	データベースのキャッシュサイズ (バイト単位) を返します。
public String	getConnectionString() [23 ページ]	SetConnectionString メソッドで登録された接続文字列を取得します。
public String	getCreationString() [23 ページ]	SetCreationString メソッドで登録された作成文字列を取得します。
public String	getEncryptionKey() [24 ページ]	setEncryptionKey メソッドで登録されたデータベース暗号化キーを取得します。
public String	getUserName() [24 ページ]	setUserName メソッドで設定されたユーザの名前を取得します。
public ConfigPersistent	setCacheSize(int) [25 ページ]	データベースのキャッシュサイズ (バイト単位) を設定します。
public void	setConnectionString(String) [26 ページ]	データベースの作成または接続に使用される接続文字列を設定します。
public void	setCreationString(String) [26 ページ]	データベースを作成するために使用される作成文字列を設定します。
public void	setEncryptionKey(String) [27 ページ]	暗号化キーを設定します。
public void	setUserName(String) [27 ページ]	ユーザの名前を設定します。

Configuration から継承されたメンバー

変更子とタイプ	メンバー	説明
public String	getDatabaseName() [29 ページ]	データベース名を返します。

変更子とタイプ	メンバー	説明
public int	getPageSize() [29 ページ]	データベースのページサイズ (バイト単位) を返します。
public Configuration	setDatabaseName(String) [29 ページ]	データベース名を設定します。
public Configuration	setPageSize(int) [30 ページ]	データベースのページサイズを設定します。
public Configuration	setPassword(String) [30 ページ]	データベースのパスワードを設定します。

このセクションの内容:

1.4.1 enableAesDBEncryption() メソッド

データベースの AES 暗号化を可能にします。

構文

```
public void enableAesDBEncryption ()
```

備考

データベースの作成時またはデータベースへの接続時に DBKEY 接続パラメータを指定するか、setEncryptionKey メソッドを使用します。

関連情報

setEncryptionKey(String) メソッド [27 ページ]

1.4.2 enableObfuscation() メソッド

データベースの難読化を有効にします。

構文

```
public void enableObfuscation ()
```

1.4.3 getCacheSize() メソッド

データベースのキャッシュサイズ (バイト単位) を返します。

構文

```
public int getCacheSize ()
```

戻り値

キャッシュサイズ。

関連情報

setCacheSize(int) メソッド [25 ページ]

1.4.4 getConnectionString() メソッド

SetConnectionString メソッドで登録された接続文字列を取得します。

構文

```
public String getConnectionString ()
```

戻り値

SetConnectionString メソッドで登録された接続文字列。

関連情報

setConnectionString(String) メソッド [26 ページ]

1.4.5 getCreationString() メソッド

SetCreationString メソッドで登録された作成文字列を取得します。

構文

```
public String getCreationString ()
```

戻り値

SetCreationString メソッドで登録された作成文字列。

関連情報

setCreationString(String) メソッド [26 ページ]

1.4.6 getEncryptionKey() メソッド

setEncryptionKey メソッドで登録されたデータベース暗号化キーを取得します。

構文

```
public String getEncryptionKey ()
```

戻り値

setEncryptionKey メソッドで登録されたデータベース暗号化キー

関連情報

setEncryptionKey(String) メソッド [27 ページ]
changeEncryptionKey(String) メソッド [39 ページ]

1.4.7 getUserName() メソッド

setUserName メソッドで設定されたユーザの名前を取得します。

構文

```
public String getUserName ()
```

戻り値

setUserName メソッドで設定されたユーザの名前。

関連情報

setUserName(String) メソッド [27 ページ]

1.4.8 setCacheSize(int) メソッド

データベースのキャッシュサイズ (バイト単位) を設定します。

構文

```
public ConfigPersistent setCacheSize (int cache_size) throws ULjException
```

パラメータ

cache_size キャッシュサイズ。すべてのプラットフォームで、デフォルトのキャッシュサイズは 20480 (20 KB) です。

戻り値

キャッシュサイズが指定された ConfigPersistent オブジェクト。

備考

キャッシュサイズによって、ページキャッシュに常駐するデータベースのページ数が決まります。サイズを拡大すると、データベースページの読み込みと書き込みの回数が減りますが、キャッシュ内でページを検索する時間が長くなります。

関連情報

getCacheSize() メソッド [22 ページ]

1.4.9 setConnectionString(String) メソッド

データベースの作成または接続に使用される接続文字列を設定します。

構文

```
public void setConnectionString (String connection_string)
```

パラメータ

connection_string データベース接続および作成で使用される接続文字列。

備考

この設定に設定されている他の項目もすべて、データベースの作成または接続のために渡されます。

1.4.10 setCreationString(String) メソッド

データベースを作成するために使用される作成文字列を設定します。

構文

```
public void setCreationString (String creation_string)
```

パラメータ

creation_string データベースの作成に使用される作成文字列。

備考

この設定に設定されている他の項目もすべて、データベースの作成のために渡されます。

1.4.11 setEncryptionKey(String) メソッド

暗号化キーを設定します。

構文

```
public void setEncryptionKey (String encryption_key)
```

パラメータ

encryption_key 暗号化キーに使用する文字列。

関連情報

getEncryptionKey() メソッド [24 ページ]
enableAesDBEncryption() メソッド [22 ページ]
changeEncryptionKey(String) メソッド [39 ページ]

1.4.12 setUserName(String) メソッド

ユーザの名前を設定します。

構文

```
public void setUserName (String user_name)
```

パラメータ

user_name ユーザの名前。

備考

この名前は、接続文字列に UID= 句を使用してネイティブデータベースに接続、またはネイティブデータベースを作成する場合に使用されます。

1.5 Configuration インタフェース

データベース用の Configuration オブジェクトを確立します。

構文

```
public interface Configuration
```

メンバー

Configuration のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public String	getDatabaseName() [29 ページ]	データベース名を返します。
public int	getPageSize() [29 ページ]	データベースのページサイズ (バイト単位) を返します。
public Configuration	setDatabaseName(String) [29 ページ]	データベース名を設定します。
public Configuration	setPageSize(int) [30 ページ]	データベースのページサイズを設定します。
public Configuration	setPassword(String) [30 ページ]	データベースのパスワードを設定します。

備考

一部の属性は、データベースの作成時にのみ使用されます。その他の属性は、データベースへの最初の接続に適用されます。データベースの作成後、またはデータベースへの接続後に設定された属性は無視されます。

このセクションの内容:

getDatabaseName() メソッド [29 ページ]
　データベース名を返します。

getPageSize() メソッド [29 ページ]
　データベースのページサイズ (バイト単位) を返します。

setDatabaseName(String) メソッド [29 ページ]
　データベース名を設定します。

setPageSize(int) メソッド [30 ページ]
　データベースのページサイズを設定します。

setPassword(String) メソッド [30 ページ]
　データベースのパスワードを設定します。

1.5.1 getDatabaseName() メソッド

データベース名を返します。

構文

```
public String getDatabaseName ()
```

戻り値

データベースの名前。

1.5.2 getPageSize() メソッド

データベースのページサイズ (バイト単位) を返します。

構文

```
public int getPageSize ()
```

戻り値

ページサイズ。

1.5.3 setDatabaseName(String) メソッド

データベース名を設定します。

構文

```
public Configuration setDatabaseName (String db_name) throws ULjException
```

パラメータ

db_name データベースの名前。

戻り値

データベース名が指定された Configuration オブジェクト。

1.5.4 setPageSize(int) メソッド

データベースのページサイズを設定します。

構文

```
public Configuration setPageSize (int page_size) throws ULjException
```

パラメータ

page_size ページサイズ（バイト単位）

戻り値

ページサイズが指定された Configuration オブジェクト。

備考

ページサイズの設定を使用して、永続的なデータベースに格納されるローの最大サイズが決定されます。また、このページサイズによって、インデックスページのサイズが確立され、各ページの子の数が決まります。

既存のデータベースを使用する場合は、データベースの作成時のページサイズにすでに設定されています。このメソッドを使用して、既存のデータベースのページサイズをリセットすることはできません。

ページサイズは、1024、2048、4096、8192、または 16384 バイトです。デフォルトは 4096 バイトです。

1.5.5 setPassword(String) メソッド

データベースのパスワードを設定します。

構文

```
public Configuration setPassword (String password) throws ULjException
```

パラメータ

password 新しいデータベースのパスワード、または既存のデータベースにアクセスするためのパスワード。

戻り値

データベースパスワードが設定された Configuration オブジェクト。

備考

このパスワードを使用してデータベースへのアクセスが許可されます。このパスワードは、データベースの作成時に指定されたパスワードと一致する必要があります。デフォルトは "dba" です。

1.6 Connection インタフェース

データベース接続を表します。データベース操作を開始するには接続が必要です。

構文

```
public interface Connection
```

メンバー

Connection のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

変数

変更子とタイプ	変数	説明
public static final byte	NOT_CONNECTED	接続されていない状態を示します。
public static final byte	CONNECTED	接続されている状態を示します。
public static final String	OPTION_DATABASE_ID	データベースオプション: データベース ID。 デフォルト値は指定されません。明示的に割り当てる必要があります。

変更子とタイプ	変数	説明
public static final String	OPTION_DATE_FORMAT	データベースオプション: 日付形式。 ConfigPersistent.setCreationString メソッドのデータベース作成文字列のみにこのオプションを設定します。 対応するオプションのデフォルト値は "YYYY-MM-DD" です。
public static final String	OPTION_DATE_ORDER	データベースオプション: 日付順。 このオプションをデータベース作成文字列の中で ConfigPersistent.setCreationString メソッドにのみ設定します。 対応するオプションのデフォルト値は "YMD" です。
public static final String	OPTION_MAX_HASH_SIZE	データベースオプション: 最大ハッシュサイズ。 このオプションをデータベース作成文字列の中で ConfigPersistent.setCreationString メソッドにのみ設定します。このオプションの名前は max_hash_size です。 対応するオプションのデフォルト値は "4" です。 インデックスを作成する SQL 文でハッシュサイズが指定されないと、デフォルトとしてこのオプションの指定値が使用されます。
public static final String	OPTION_NEAREST_CENTURY	データベースオプション: 基準年。 このオプションをデータベース作成文字列の中で ConfigPersistent.setCreationString メソッドにのみ設定します。 対応するオプションのデフォルト値は "50" です。
public static final String	OPTION_PRECISION	データベースオプション: 精度。 このオプションをデータベース作成文字列の中で ConfigPersistent.setCreationString メソッドにのみ設定します。 対応するオプションのデフォルト値は "30" です。

変更子とタイプ	変数	説明
public static final String	OPTION_TIME_FORMAT	データベースオプション: 時間形式。 このオプションをデータベース作成文字列の中で ConfigPersistent.setCreationString メソッドにのみ設定します。 対応するオプションのデフォルト値は "HH:NN:SS.SSS" です。
public static final String	OPTION_TIMESTAMP_FORMAT	データベースオプション: タイムスタンプ形式。 このオプションをデータベース作成文字列の中で ConfigPersistent.setCreationString メソッドにのみ設定します。 対応するオプションのデフォルト値は "YYYY-MM-DD HH:NN:SS.SSS" です。
public static final String	OPTION_TIMESTAMP_WITH_TIME_ZONE_FORMAT	データベースオプション: タイムゾーン形式のタイムスタンプ。 このオプションをデータベース作成文字列の中で ConfigPersistent.setCreationString メソッドにのみ設定します。 対応するオプションのデフォルト値は "YYYY-MM-DD HH:NN:SS.SSS+HH:NN" です。
public static final String	OPTION_TIMESTAMP_INCREMENT	データベースオプション: タイムスタンプインクリメント。 このオプションをデータベース作成文字列の中で ConfigPersistent.setCreationString メソッドにのみ設定します。 対応するオプションのデフォルト値は "1" です。
public static final String	OPTION_SCALE	データベースオプション: 位取り。 このオプションをデータベース作成文字列の中で ConfigPersistent.setCreationString メソッドにのみ設定します。 対応するオプションのデフォルト値は "6" です。
public static final String	OPTION_ML_REMOTE_ID	データベースオプション: ML リモート ID。 デフォルト値は指定されません。最初の Mobile Link 同期の後で値が設定されます。

変更子とタイプ	変数	説明
public static final String	PROPERTY_DATABASE_NAME	データベースプロパティ: データベース名。 Configuration.setDatabaseName メソッドでこのプロパティを設定します。
public static final String	PROPERTY_PAGE_SIZE	データベースプロパティ: ページサイズ。 Configuration.setPageSize メソッドでこのプロパティを設定します。
public static final String	SYNC_ALL	データベース内の全テーブルの同期を要求するために使用するパブリケーションのリストです。どのパブリケーションにも含まれないテーブルも含まれます。 NoSync と指定されているテーブルは同期されません。 この定数は、NULL 参照または空の文字列と同じです。
public static final String	SYNC_ALL_PUBS	データベース内の全パブリケーションの同期を要求するために使用するパブリケーションのリストです。 NoSync と指定されているテーブルは同期されません。
public static final String	SYNC_ALL_DB_PUB_NAME	SYNC_ALL_DB パブリケーションの予約名です。
public static final int	ULVF_TABLE	テーブルを検証するために使用します。 テーブルとインデックスのローカウントが一致しているかどうかをチェックします。
public static final int	ULVF_INDEX	インデックスを検証するために使用します。 インデックスの整合性をチェックします。
public static final int	ULVF_DATABASE	データベースを検証するために使用します。 ページのチェックサムやその他のチェックを使用してデータベースページを検証します。
public static final int	ULVF_EXPRESS	完全ではないが、高速な検証を実行するために使用します。 このフラグは他のフラグと組み合わせることで動作を変更させます。
public static final int	ULVF_FULL_VALIDATE	データベース上ですべてのタイプの検証を実行します。

メソッド

変更子とタイプ	メソッド	説明
public void	cancelWaitForEvent() [39 ページ]	この Connection オブジェクトの waitForEvent 呼び出しをすべてキャンセルします。
public void	changeEncryptionKey(String) [39 ページ]	Ultra Light データベースのデータベース暗号化キーを変更します。
public void	commit() [40 ページ]	データベースの変更内容をコミットします。
public DecimalNumber	createDecimalNumber [40 ページ]	新しい DecimalNumber オブジェクトを作成します。
public SyncParms	createSyncParms [42 ページ]	同期パラメータセットを作成します。
public UUIDValue	createUUIDValue() [44 ページ]	UUID 値を作成します。
public void	dropDatabase() [44 ページ]	データベースを削除します。
public DatabaseInfo	getDatabaseInfo() [45 ページ]	データベースプロパティに関する情報を含む DataInfo オブジェクトを返します。
public String	getDatabaseProperty(String) [45 ページ]	データベースプロパティを返します。
public Date	getLastDownloadTime(String) [46 ページ]	指定されたパブリケーションの最後のダウンロードの時刻を返します。
public long	getLastIdentity() [46 ページ]	DEFAULT AUTOINCREMENT カラムまたは DEFAULT GLOBAL AUTOINCREMENT カラムに挿入された最新の値が取得されます。最新の INSERT トランザクションが、このようなカラムがないテーブルに対して行われた場合は 0 になります。
public SQLInfo	getLastWarning() [47 ページ]	この接続で最後に実行された SQL 文に関する情報を返します。
public String	getOption(String) [47 ページ]	データベースオプションを返します。
public byte	getState() [48 ページ]	接続のステータスを返します。
public SyncObserver	getSyncObserver() [49 ページ]	この Connection オブジェクトに対して現在登録されている SyncObserver オブジェクトを返します。
public SyncResult	getSyncResult() [49 ページ]	前回の SYNCHRONIZE SQL 文の結果を返します。
public PreparedStatement	prepareStatement(String) [50 ページ]	実行する文を準備します。
public void	registerForEvent(short, String) [51 ページ]	システムイベントを登録して通知を受信します。
public void	release() [51 ページ]	この接続を解放します。
public void	resetLastDownloadTime(String) [52 ページ]	指定されたパブリケーションのダウンロード時刻をリセットします。

変更子とタイプ	メソッド	説明
public void	rollback() [52 ページ]	データベースへの変更を取り消すロールバックをコミットします。
public void	rollbackPartialDownload() [53 ページ]	失敗した同期からの変更をロールバックします。
public void	setDatabaseId(int) [53 ページ]	グローバルオートインクリメントのデータベース ID を設定します。
public void	setOption(String, String) [54 ページ]	データベースオプションを設定します。
public void	setSyncObserver(SyncObserver) [54 ページ]	この接続で同期の進行状況をモニタする SyncObserver オブジェクトを設定します。
public void	synchronize(SyncParms) [55 ページ]	データベースを Mobile Link サーバと同期させます。
public void	unregisterForEvent(short, String) [56 ページ]	システムイベントの登録を解除して通知の受信を停止します。
public void	validateDatabase(int, ValidateDatabaseProgressListener, String) [56 ページ]	この接続でのデータベースを検証します。
public ULjEvent	waitForEvent(int) [57 ページ]	イベント通知が発生するまで待ちます。

備考

接続は、DatabaseManager クラスの connect メソッドまたは createDatabase メソッドを使用して取得します。接続が不要になったら release メソッドを使用します。データベースのすべての接続を解放したら、データベースは終了します。

Connection オブジェクトには、次の機能があります。

- 新しいスキーマの作成 (テーブル、インデックス、パブリケーション)
- 新しい値とドメインオブジェクトの作成
- データベースへの変更の永続的なコミット
- 実行する SQL 文の準備
- コミットされていないデータベースへの変更のロールバック

次の例は、単純なデータベースのために作成された Connection オブジェクト conn を使用して、このデータベースのスキーマを作成する方法を示しています。データベースにはテーブル T1 と T2 があります。T1 には num という整数のプライマリキーカラムが 1 つあります。T2 には num という整数のプライマリキーカラムと quantity という整数カラムがあります。T2 の quantity には追加インデックスがあります。T1 は PubA というパブリケーションに含まれます。

```
// Assumes a valid connection object, conn, for the current database.
PreparedStatement ps;
ps = conn.prepareStatement( "CREATE TABLE T1 ( num INT NOT NULL PRIMARY KEY )" );
ps.execute();
ps.close();
ps = conn.prepareStatement( "CREATE TABLE T2 ( num INT NOT NULL PRIMARY KEY,
quantity INT)" );
ps.execute();
ps.close();
ps = conn.prepareStatement( "CREATE INDEX index1 ON T2( quantity )" );
```

```
ps.execute();
ps.close();
ps = conn.prepareStatement( "CREATE Publication PubA ( Table T1 )" );
ps.execute();
ps.close();
```

このセクションの内容:

prepareStatement(String) メソッド [50 ページ]
実行する文を準備します。

registerForEvent(short, String) メソッド [51 ページ]
システムイベントを登録して通知を受信します。

release() メソッド [51 ページ]
この接続を解放します。

resetLastDownloadTime(String) メソッド [52 ページ]
指定されたパブリケーションのダウンロード時刻をリセットします。

rollback() メソッド [52 ページ]
データベースへの変更を取り消すロールバックをコミットします。

rollbackPartialDownload() メソッド [53 ページ]
失敗した同期からの変更をロールバックします。

setDatabaseId(int) メソッド [53 ページ]
グローバルオートインクリメントのデータベース ID を設定します。

setOption(String, String) メソッド [54 ページ]
データベースオプションを設定します。

setSyncObserver(SyncObserver) メソッド [54 ページ]
この接続で同期の進行状況をモニタする SyncObserver オブジェクトを設定します。

synchronize(SyncParms) メソッド [55 ページ]
データベースを Mobile Link サーバと同期させます。

unregisterForEvent(short, String) メソッド [56 ページ]
システムイベントの登録を解除して通知の受信を停止します。

validateDatabase(int, ValidateDatabaseProgressListener, String) メソッド [56 ページ]
この接続でのデータベースを検証します。

waitForEvent(int) メソッド [57 ページ]
イベント通知が発生するまで待ちます。

関連情報

DatabaseManager クラス [62 ページ]
createDatabase(Configuration) メソッド [66 ページ]
connect(Configuration) メソッド [64 ページ]
release() メソッド [51 ページ]

1.6.1 cancelWaitForEvent() メソッド

この Connection オブジェクトの waitForEvent 呼び出しをすべてキャンセルします。

構文

```
public void cancelWaitForEvent () throws ULjException
```

関連情報

waitForEvent(int) メソッド [57 ページ]

1.6.2 changeEncryptionKey(String) メソッド

Ultra Light データベースのデータベース暗号化キーを変更します。

構文

```
public void changeEncryptionKey (String newKey) throws ULjException
```

パラメータ

newKey データベースの新しい暗号化キー。

備考

このメソッドを呼び出すアプリケーションでは、データベースが同期されていること、または信頼できるバックアップコピーが作成されていることを、先に確認しておく必要があります。changeEncryptionKey メソッドは、完了させることが必要な操作であるため、信頼できるデータベース バックアップを作成しておくことが重要です。データベース暗号化キーを変更すると、まずデータベースのすべてのローは古いキーを使用して復号され、次に新しいキーを使用して再度暗号化されて、書き込まれます。この操作は元に戻せません。暗号化変更処理が完了しなかった場合、データベースは無効な状態のままになり、再度アクセスすることはできなくなります。

1.6.3 commit() メソッド

データベースの変更内容をコミットします。

構文

```
public void commit () throws ULjException
```

備考

このメソッドを呼び出すと、最後のコミット後またはロールバック後に行われたテーブルデータへの変更がすべて永続的になります。

1.6.4 createDecimalNumber メソッド

新しい DecimalNumber オブジェクトを作成します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public DecimalNumber	createDecimalNumber(int, int) [41 ページ]	DecimalNumber オブジェクトを作成します。
public DecimalNumber	createDecimalNumber(int, int, String) [41 ページ]	DecimalNumber オブジェクトを作成します。

このセクションの内容:

createDecimalNumber(int, int) メソッド [41 ページ]
DecimalNumber オブジェクトを作成します。

createDecimalNumber(int, int, String) メソッド [41 ページ]
DecimalNumber オブジェクトを作成します。

1.6.4.1 createDecimalNumber(int, int) メソッド

DecimalNumber オブジェクトを作成します。

構文

```
public DecimalNumber createDecimalNumber (
    int precision,
    int scale
) throws ULjException
```

パラメータ

precision 数値の桁数。
scale 数値の小数点以下の桁数。

戻り値

指定された型の DecimalNumber オブジェクト。

関連情報

DecimalNumber インタフェース [69 ページ]

1.6.4.2 createDecimalNumber(int, int, String) メソッド

DecimalNumber オブジェクトを作成します。

構文

```
public DecimalNumber createDecimalNumber (
    int precision,
    int scale,
    String value
) throws ULjException
```

パラメータ

precision 数値の桁数。
scale 数値の小数点以下の桁数。
value 設定する値。

戻り値

指定された型の DecimalNumber オブジェクト。

関連情報

DecimalNumber インタフェース [69 ページ]

1.6.5 createSyncParms メソッド

同期パラメータセットを作成します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public SyncParms	createSyncParms(String, String) [43 ページ]	HTTP 同期用の同期パラメータセットを作成します。
public SyncParms	createSyncParms(int, String, String) [43 ページ]	HTTP 同期用の同期パラメータセットを作成します。

このセクションの内容:

createSyncParms(String, String) メソッド [43 ページ]
HTTP 同期用の同期パラメータセットを作成します。

createSyncParms(int, String, String) メソッド [43 ページ]
HTTP 同期用の同期パラメータセットを作成します。

1.6.5.1 createSyncParms(String, String) メソッド

HTTP 同期用の同期パラメータセットを作成します。

構文

```
public SyncParms createSyncParms (
    String userName,
    String version
) throws ULjException
```

パラメータ

userName このクライアントデータベース用のユニークな Mobile Link ユーザ名。
version Mobile Link スクリプトのバージョン。

戻り値

SyncParms オブジェクト。

関連情報

createSyncParms(int, String, String) メソッド [43 ページ]
setUserName(String) メソッド [223 ページ]

1.6.5.2 createSyncParms(int, String, String) メソッド

HTTP 同期用の同期パラメータセットを作成します。

構文

```
public SyncParms createSyncParms (
    int streamType,
    String userName,
    String version
) throws ULjException
```

パラメータ

streamType 同期ストリームのタイプの指定に使用する、SyncParms クラス内で定義されている定数の 1 つ。
userName Mobile Link ユーザ名
version Mobile Link スクリプトのバージョン。

戻り値

SyncParms オブジェクト。

関連情報

createSyncParms(String, String) メソッド [43 ページ]

1.6.6 createUUIDValue() メソッド

UUID 値を作成します。

構文

```
public UUIDValue createUUIDValue () throws ULjException
```

戻り値

ドメインの UUIDValue インスタンス。

1.6.7 dropDatabase() メソッド

データベースを削除します。

構文

```
public void dropDatabase () throws ULjException
```

備考

接続によって参照されているデータベースを消去し、接続を解放します。削除するデータベースへの有効な接続が、この接続だけである必要があります。

1.6.8 getDatabaseInfo() メソッド

データベースプロパティに関する情報を含む DataInfo オブジェクトを返します。

構文

```
public DatabaseInfo getDatabaseInfo () throws ULjException
```

戻り値

DatabaseInfo オブジェクト。

1.6.9 getDatabaseProperty(String) メソッド

データベースプロパティを返します。

構文

```
public String getDatabaseProperty (String name) throws ULjException
```

パラメータ

name データベースプロパティの名前。このパラメータにサポート対象の任意の Ultra Light データベースプロパティ名を設定できます。

戻り値

指定された名前に対応するプロパティの値。

1.6.10 getLastDownloadTime(String) メソッド

指定されたパブリケーションの最後のダウンロードの時刻を返します。

構文

```
public Date getLastDownloadTime (String pub_name) throws ULjException
```

パラメータ

pub_name チェックするパブリケーションの名前。このパラメータは、1 つのパブリケーションを参照するか、データベース全体を最後にダウンロードしたときの特殊な Connection.SYNC_ALL_DB_PUB_NAME パブリケーションである必要があります。

戻り値

最後のダウンロードのタイムスタンプ。

関連情報

resetLastDownloadTime(String) メソッド [52 ページ]

1.6.11 getLastIdentity() メソッド

DEFAULT AUTOINCREMENT カラムまたは DEFAULT GLOBAL AUTOINCREMENT カラムに挿入された最新の値が取得されます。最新の INSERT トランザクションが、このようなカラムがないテーブルに対して行われた場合は 0 になります。

構文

```
public long getLastIdentity ()
```

戻り値

直前に使用した identity の値。

備考

テーブルに複数の (GLOBAL) AUTOINCREMENT 型のカラムが含まれている場合、この値が属しているカラムは特定されません。

1.6.12 getLastWarning() メソッド

この接続で最後に実行された SQL 文に関する情報を返します。

構文

```
public SQLInfo getLastWarning ()
```

戻り値

最後に実行された SQL 文の SQLInfo

1.6.13 getOption(String) メソッド

データベースオプションを返します。

構文

```
public String getOption (String option_name) throws ULjException
```

パラメータ

option_name 取得するオプションの名前。

option_name 取得するオプションの名前。このパラメータには、接続インタフェース内の *OPTION_* プレフィクスが付いている任意の定数を設定できます。

戻り値

データベースオプションの値。

備考

データベースオプションはデータベース内に格納され、オプションの設定後にデータベースに接続したときにも取得できます。

データベースの作成時に、一連の必須オプションが作成されます。

関連情報

setOption(String, String) メソッド [54 ページ]

1.6.14 getState() メソッド

接続のステータスを返します。

構文

```
public byte getState () throws ULjException
```

戻り値

接続のステータスを示すバイト数。

備考

次の例は、接続状態を確認し、接続を解放する方法を示しています。

```
if( _conn.getState() == Connection.CONNECTED ){
   _conn.release();
}
```

関連情報

Connection インタフェース [31 ページ]

1.6.15 getSyncObserver() メソッド

この Connection オブジェクトに対して現在登録されている SyncObserver オブジェクトを返します。

構文

```
public SyncObserver getSyncObserver ()
```

戻り値

SyncObserver、または observer が存在しない場合は NULL。

関連情報

setSyncObserver(SyncObserver) メソッド [54 ページ]

1.6.16 getSyncResult() メソッド

前回の SYNCHRONIZE SQL 文の結果を返します。

構文

```
public SyncResult getSyncResult ()
```

戻り値

前回の SYNCHRONIZE SQL 文の結果を示す SyncResult オブジェクト。

備考

次に、前回の SYNCHRONIZE SQL 文の結果を取得する例を示します。

```
PreparedStatement ps = conn.prepareStatement("SYNCHRONIZE PROFILE myprofile");
ps.execute();
ps.close();
SyncResult result = conn.getSyncResult();
```

```
display(
   "*** Synchronized *** sent=" + result.getSentRowCount()
   + ", received=" + result.getReceivedRowCount()
);
```

注記

このメソッドでは、前回の Connection.synchronize メソッド呼び出しの結果は返されません。前回の Connection.synchronize(SyncParms) メソッド呼び出しの SyncResult オブジェクトを取得するには、渡された SyncParms オブジェクトで getSyncResult メソッドを使用します。

関連情報

SyncResult クラス [224 ページ]
getSyncResult() メソッド [208 ページ]

1.6.17 prepareStatement(String) メソッド

実行する文を準備します。

構文

```
public PreparedStatement prepareStatement (String sql) throws ULjException
```

パラメータ

sql 準備する SQL 文

戻り値

PreparedStatement オブジェクト。

関連情報

PreparedStatement インタフェース [106 ページ]

1.6.18 registerForEvent(short, String) メソッド

システムイベントを登録して通知を受信します。

構文

```
public void registerForEvent (
    short event_type,
    String object_name
) throws ULjException
```

パラメータ

event_type 登録するイベントのタイプ

object_name イベントを適用するオブジェクト (テーブル名など)。

関連情報

ULjEvent インタフェース [242 ページ]

1.6.19 release() メソッド

この接続を解放します。

構文

```
public void release () throws ULjException
```

備考

一度解放した接続は、データベースへのアクセスに使用できなくなります。

コミットされていないトランザクションがある接続を解放しようとするとエラーが発生します。

1.6.20 resetLastDownloadTime(String) メソッド

指定されたパブリケーションのダウンロード時刻をリセットします。

構文

```
public void resetLastDownloadTime (String pub_name) throws ULjException
```

パラメータ

pub_name チェックするパブリケーションの名前。

備考

データベース全体が同期されたダウンロード時刻をリセットするには、特殊なパブリケーション Connection.SYNC_ALL_DB_PUB_NAME を使用します。

このメソッドを使用するには、現在の接続に、コミットされていないトランザクションがないことが必要です。

1.6.21 rollback() メソッド

データベースへの変更を取り消すロールバックをコミットします。

構文

```
public void rollback () throws ULjException
```

備考

このメソッドを呼び出すと、この Connection オブジェクトで、最後のコミット後またはロールバック後に行われたテーブルデータへの変更がすべて取り消されます。

1.6.22 rollbackPartialDownload() メソッド

失敗した同期からの変更をロールバックします。

構文

```
public void rollbackPartialDownload () throws ULjException
```

備考

このメソッドが影響するのは、再開可能なダウンロードだけです(SyncParms.setKeepPartialDownload を true に設定して同期を実行した場合)。

KeepPartialDownload パラメータが true に設定されていて、同期のダウンロード段階で通信エラーが発生した場合、ダウンロード済みの変更が保持され、ダウンロードが中断された位置から同期を再開できます。

ダウンロードの再開が必要ない場合は、このメソッドで部分ダウンロードを削除します。

関連情報

setKeepPartialDownload(boolean) メソッド [215 ページ]

1.6.23 setDatabaseId(int) メソッド

グローバルオートインクリメントのデータベース ID を設定します。

構文

```
public void setDatabaseId (int id) throws ULjException
```

パラメータ

id データベース ID。

備考

データベース ID にはデフォルト値はありません。

データベース ID を明示的に設定しないかぎり、INSERT 時に GLOBAL AUTOINCREMENT カラムに NULL 値が挿入されます。

1.6.24 setOption(String, String) メソッド

データベースオプションを設定します。

構文

```
public void setOption (
    String option_name,
    String option_value
) throws ULjException
```

パラメータ

option_name 設定するオプションの名前。このパラメータにサポート対象の任意の Ultra Light データベースオプション名を設定できます。
option_value オプションの新しい値。

備考

オプションが現在データベースに格納されていない場合は作成されます。

この接続にコミットされていないトランザクションがあるときに、このメソッドを呼び出すことはできません。

1.6.25 setSyncObserver(SyncObserver) メソッド

この接続で同期の進行状況をモニタする SyncObserver オブジェクトを設定します。

構文

```
public void setSyncObserver (SyncObserver so)
```

パラメータ

so SyncObserver オブジェクト、または現在登録されている SyncObserver オブジェクトを削除する場合はヌル。

備考

この SyncObserver オブジェクトは、以降の SYNCHRONIZE SQL 文に使用されます。

デフォルトは NULL で、これは observer なしを示します。

関連情報

SyncObserver インタフェース [194 ページ]

1.6.26 synchronize(SyncParms) メソッド

データベースを Mobile Link サーバと同期させます。

構文

```
public void synchronize (SyncParms config) throws ULjException
```

パラメータ

config 同期に使用するパラメータが含まれている SyncParms オブジェクト。

関連情報

SyncParms クラス [198 ページ]

1.6.27 unregisterForEvent(short, String) メソッド

システムイベントの登録を解除して通知の受信を停止します。

構文

```
public void unregisterForEvent (
    short event_type,
    String object_name
) throws ULjException
```

パラメータ

event_type 登録を解除するイベントのタイプ。

object_name イベントを適用するオブジェクト (テーブル名など)。

関連情報

ULjEvent インタフェース [242 ページ]

1.6.28 validateDatabase(int, ValidateDatabaseProgressListener, String) メソッド

この接続でのデータベースを検証します。

構文

```
public void validateDatabase (
    int flags,
    ValidateDatabaseProgressListener listener,
    String tableName
) throws ULjException
```

パラメータ

flags 検証のタイプを制御するフラグ。

listener 検証の進行状況の情報を受け取るリスナ。

tableName 検証する特定のテーブル。すべてのテーブルの場合は NULL。

備考

このルーチンに渡されるフラグに応じて、テーブル、インデックス、およびデータベースページを検証できます。検証中に情報を受け取るには、コールバック関数を実装し、アドレスをこのルーチンに渡します。検証対象を特定のテーブルに限定するには、テーブルの名前または ID を最後のパラメータとして渡します。

flags パラメータは、次のいずれかの値の組み合わせです。

- ULVF_TABLE
- ULVF_INDEX
- ULVF_DATABASE
- ULVF_EXPRESS
- ULVF_FULL_VALIDATE

次の例は、エクスプレスモードでのテーブルとインデックスの検証を示します。

```
flags = ULVF_TABLE | ULVF_INDEX | ULVF_EXPRESS;
```

1.6.29 waitForEvent(int) メソッド

イベント通知が発生するまで待ちます。

構文

```
public ULjEvent waitForEvent (int wait_ms) throws ULjException
```

パラメータ

wait_ms 返す前に、待機 (ブロック) する時間 (ミリ秒単位)。無限に待機する場合は、-1 に設定します。

戻り値

待機期間内に発生したイベント。待機期間内に通知を受信しなかった場合は NULL。

備考

この呼び出しは、通知が受信されるまで、または指定された待機時間が経過するまでブロックします。待機を取り消す場合は、cancelWaitForEvent メソッドを使用します。

関連情報

ULjEvent インタフェース [242 ページ]
cancelWaitForEvent() メソッド [39 ページ]

1.7 DatabaseInfo インタフェース

Connection オブジェクトに関連付けられ、データベース情報を公開するメソッドを提供します。

構文

```
public interface DatabaseInfo
```

メンバー

DatabaseInfo のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public int	getNumberRowsToUpload [59 ページ]	アップロードを待機しているローの数を返します。
public int	getPageReads() [61 ページ]	ページの読み込み数を返します。
public int	getPageSize() [61 ページ]	データベースのページサイズ (バイト単位) を返します。
public int	getPageWrites() [61 ページ]	ページの書き込み数を返します。
public String	getRelease() [62 ページ]	ソフトウェアのリリース番号を返します。

備考

このインタフェースは、Connection オブジェクトの getDatabaseInfo メソッドを使用して呼び出します。

このセクションの内容:

getNumberRowsToUpload メソッド [59 ページ]
アップロードを待機しているローの数を返します。

getPageReads() メソッド [61 ページ]
ページの読み込み数を返します。

getPageSize() メソッド [61 ページ]
データベースのページサイズ (バイト単位) を返します。

getPageWrites() メソッド [61 ページ]
ページの書き込み数を返します。

getRelease() メソッド [62 ページ]
ソフトウェアのリリース番号を返します。

関連情報

Connection インタフェース [31 ページ]
getDatabaseInfo() メソッド [45 ページ]

1.7.1 getNumberRowsToUpload メソッド

アップロードを待機しているローの数を返します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public int	getNumberRowsToUpload() [60 ページ]	アップロードを待機しているローの数を返します。
public int	getNumberRowsToUpload(String, int) [60 ページ]	指定したスレッショルドまでアップロードを待機しているローの数を返します。

このセクションの内容:

getNumberRowsToUpload() メソッド [60 ページ]
アップロードを待機しているローの数を返します。

getNumberRowsToUpload(String, int) メソッド [60 ページ]
指定したスレッショルドまでアップロードを待機しているローの数を返します。

1.7.1.1 getNumberRowsToUpload() メソッド

アップロードを待機しているローの数を返します。

構文

```
public int getNumberRowsToUpload ()
```

戻り値

ローの数。

1.7.1.2 getNumberRowsToUpload(String, int) メソッド

指定したスレッショルドまでアップロードを待機しているローの数を返します。

構文

```
public int getNumberRowsToUpload (
    String pubList,
    int threshold
)
```

パラメータ

pubList チェック対象となるパブリケーションのカンマ区切りのリストを含む文字列。空の文字列 (UL_SYNC_ALL マクロ) は、非同期とマーク付けされたものを除くすべてのテーブルを表します。アスタリスクのみの文字列 (UL_SYNC_ALL_PUBS マクロ) は、いずれかのパブリケーションで参照されているすべてのテーブルを表します。一部のテーブルは、どのパブリケーションの一部でもないため、この値が * の場合は含まれません。

threshold カウントするローの最大数。この呼び出しの所要時間を制限します。threshold が 0 の場合、制限はありません (つまり、同期する必要のあるすべてのローをカウントします)。threshold が 1 の場合、同期の必要なローがあるかどうかを簡単に判別するために使用できます。

戻り値

指定されたパブリケーションのセットまたはデータベース全体のいずれかで、同期を必要とするローの数。

1.7.2 getPageReads() メソッド

ページの読み込み数を返します。

構文

```
public int getPageReads ()
```

戻り値

ページ読み込み数。

備考

この数はページ読み込みの累積合計であり、このクラスのインスタンス変数に格納されます。

1.7.3 getPageSize() メソッド

データベースのページサイズ (バイト単位) を返します。

構文

```
public int getPageSize ()
```

戻り値

ページサイズ。

1.7.4 getPageWrites() メソッド

ページの書き込み数を返します。

構文

```
public int getPageWrites ()
```

戻り値

ページ書き込み数。

備考

この数はページ書き込みの累積合計であり、このクラスのインスタンス変数に格納されます。

1.7.5 getRelease() メソッド

ソフトウェアのリリース番号を返します。

構文

```
public String getRelease ()
```

戻り値

リリース番号。

備考

たとえば、ソフトウェアリリースの値 "12.0.1.1234" は、リリースが 12.0.1、ビルド番号が 1234 であることを示します。

1.8 DatabaseManager クラス

基本設定を取得したり、新しいデータベースを作成したり、既存のデータベースに接続したりするための静的メソッドを提供します。

構文

```
public class DatabaseManager
```

メンバー

DatabaseManager のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public static Connection	connect(Configuration) [64 ページ]	設定に基づいて既存のデータベースに接続します。
public static ConfigFileAndroid	createConfigurationFileAndroid(String, android.content.Context) [65 ページ]	ファイルから物理データベースストア用の Configuration オブジェクトを作成し、ConfigFileAndroid オブジェクトを返します。
public static Connection	createDatabase(Configuration) [66 ページ]	設定セットに基づいて新しいデータベースを作成し、データベースに接続します。
public static FileTransfer	createFileTransfer(String, int, String, String) [66 ページ]	Mobile Link との間でファイルを転送するための FileTransfer オブジェクトを作成します。
public static FileTransfer	createFileTransferAndroid(android.content.Context, String, int, String, String) [67 ページ]	Mobile Link との間でファイルを転送するための FileTransfer オブジェクトを作成します。
public static void	release() [68 ページ]	DatabaseManager オブジェクトを閉じてすべての接続を解放し、すべてのデータベースを停止します。

備考

次の例は、既存のデータベースを開く方法、またはデータベースが存在しない場合は新規に作成する方法を示しています。

```
Connection conn = null;
ConfigFileAndroid config = null;
try {
   config = DatabaseManager.createConfigurationFileAndroid(
        "test.udb", getApplicationContext()
   );
   conn = DatabaseManager.connect(config);
} catch(ULjException ex) {
   if (config != null) {
       try {
           conn = DatabaseManager.createDatabase(config);
           // Create the schema here.
       } catch(ULjException exception) {
           // An error has occurred.
       }
   }
}
```

このセクションの内容:

createConfigurationFileAndroid(String, android.content.Context) メソッド [65 ページ]
ファイルから物理データベースストア用の Configuration オブジェクトを作成し、ConfigFileAndroid オブジェクトを返します。

createDatabase(Configuration) メソッド [66 ページ]
設定セットに基づいて新しいデータベースを作成し、データベースに接続します。

createFileTransfer(String, int, String, String) メソッド [66 ページ]
Mobile Link との間でファイルを転送するための FileTransfer オブジェクトを作成します。

createFileTransferAndroid(android.content.Context, String, int, String, String) メソッド [67 ページ]
Mobile Link との間でファイルを転送するための FileTransfer オブジェクトを作成します。

release() メソッド [68 ページ]
DatabaseManager オブジェクトを閉じてすべての接続を解放し、すべてのデータベースを停止します。

関連情報

Connection インタフェース [31 ページ]
Configuration インタフェース [28 ページ]

1.8.1 connect(Configuration) メソッド

設定に基づいて既存のデータベースに接続します。

構文

```
public static Connection connect (Configuration config) throws ULjException
```

パラメータ

config 既存のデータベースの仕様が含まれている Configuration オブジェクト。

戻り値

データベースへの接続を確立する Connection オブジェクト。

関連情報

Configuration インタフェース [28 ページ]
Connection インタフェース [31 ページ]

1.8.2 createConfigurationFileAndroid(String, android.content.Context) メソッド

ファイルから物理データベーススストア用の Configuration オブジェクトを作成し、ConfigFileAndroid オブジェクトを返します。

構文

```
public static ConfigFileAndroid createConfigurationFileAndroid (
    String file_name,
    android.content.Context context
) throws ULjException
```

パラメータ

file_name 使用または作成するデータベースファイルの名前。データベースパスへのアクセス権があること。このファイルのデフォルトパスは、`/data/data/your-application-package-name/` です。*your-application-package-name* はアプリケーションに割り当てたパッケージ名です。ファイル名に絶対パスを含めることでデータベースの別の場所を指定できます。
context Android アプリケーションからの Context オブジェクト。このパラメータを NULL にすることはできません。

戻り値

データベースの設定に使用された ConfigFileAndroid オブジェクト。

関連情報

ConfigFileAndroid インタフェース [18 ページ]

1.8.3 createDatabase(Configuration) メソッド

設定セットに基づいて新しいデータベースを作成し、データベースに接続します。

構文

```
public static Connection createDatabase (Configuration config) throws ULjException
```

パラメータ

config 新規のデータベースの仕様が含まれている Configuration オブジェクト。

戻り値

新規のデータベースへの接続を確立する Connection オブジェクト。

備考

このメソッドは、デバイス上の同じ名前の既存データベースを置換します。

関連情報

Configuration インタフェース [28 ページ]
Connection インタフェース [31 ページ]

1.8.4 createFileTransfer(String, int, String, String) メソッド

Mobile Link との間でファイルを転送するための FileTransfer オブジェクトを作成します。

構文

```
public static FileTransfer createFileTransfer (
    String fileName,
    int streamType,
    String userName,
    String version
```

```
) throws ULjException
```

パラメータ

fileName 転送するサーバファイルの名前。このパラメータにパス情報を含めることはできません。
streamType 通信ストリームタイプの識別に使用する、SyncParms クラス内で定義されている定数の 1 つ。
userName Mobile Link ユーザ名
version Mobile Link スクリプトのバージョン。

戻り値

FileTransfer オブジェクト。

関連情報

SyncParms クラス [198 ページ]

1.8.5 createFileTransferAndroid(android.content.Context, String, int, String, String) メソッド

Mobile Link との間でファイルを転送するための FileTransfer オブジェクトを作成します。

構文

```
public static FileTransfer createFileTransferAndroid (
    android.content.Context context,
    String fileName,
    int streamType,
    String userName,
    String version
) throws ULjException
```

パラメータ

context Android アプリケーションからの Context オブジェクト。このパラメータを NULL にすることはできません。
fileName 転送するサーバファイルの名前。このパラメータにパス情報を含めることはできません。

streamType 通信ストリームタイプの識別に使用する、SyncParms クラス内で定義されている定数の 1 つ。
userName Mobile Link ユーザ名
version Mobile Link スクリプトのバージョン。

戻り値

FileTransfer オブジェクト。

備考

createConfigurationFileAndroid メソッドでデータベース接続が作成されていない場合はこのメソッドを使用する必要があります。

関連情報

createConfigurationFileAndroid(String, android.content.Context) メソッド [65 ページ]

1.8.6 release() メソッド

DatabaseManager オブジェクトを閉じてすべての接続を解放し、すべてのデータベースを停止します。

構文

```
public static void release () throws ULjException
```

備考

このメソッドは DatabaseManager で作成されたすべての接続を解放します。

コミットされていないトランザクションはロールバックされます。

1.9 DecimalNumber インタフェース

正確な decimal 値を表し、java.math.BigDecimal を使用できない Java プラットフォームに 10 進法計算のサポートを提供します。

構文

```
public interface DecimalNumber
```

メンバー

DecimalNumber のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public DecimalNumber	add(DecimalNumber, DecimalNumber) [70 ページ]	2 つの DecimalNumber オブジェクトを加算し、その和を返します。
public DecimalNumber	divide(DecimalNumber, DecimalNumber) [71 ページ]	最初の DecimalNumber オブジェクトを 2 番目の DecimalNumber オブジェクトで割って、その商を返します。
public String	getString() [71 ページ]	DecimalNumber オブジェクトの String 表現を返します。
public boolean	isNull() [72 ページ]	DecimalNumber オブジェクトが NULL かどうかを確認します。
public DecimalNumber	multiply(DecimalNumber, DecimalNumber) [72 ページ]	2 つの DecimalNumber オブジェクトを乗算し、その積を返します。
public void	set(String) [73 ページ]	DecimalNumber オブジェクトに String 値を設定します。
public void	setNull() [73 ページ]	DecimalNumber オブジェクトを NULL に設定します。
public DecimalNumber	subtract(DecimalNumber, DecimalNumber) [73 ページ]	2 番目の DecimalNumber オブジェクトを最初の DecimalNumber オブジェクトから減算し、その差を返します。

このセクションの内容:

add(DecimalNumber, DecimalNumber) メソッド [70 ページ]
2 つの DecimalNumber オブジェクトを加算し、その和を返します。

divide(DecimalNumber, DecimalNumber) メソッド [71 ページ]
最初の DecimalNumber オブジェクトを 2 番目の DecimalNumber オブジェクトで割って、その商を返します。

getString() メソッド [71 ページ]

DecimalNumber オブジェクトの String 表現を返します。

isNull() メソッド [72 ページ]
DecimalNumber オブジェクトが NULL かどうかを確認します。

multiply(DecimalNumber, DecimalNumber) メソッド [72 ページ]
2 つの DecimalNumber オブジェクトを乗算し、その積を返します。

set(String) メソッド [73 ページ]
DecimalNumber オブジェクトに String 値を設定します。

setNull() メソッド [73 ページ]
DecimalNumber オブジェクトを NULL に設定します。

subtract(DecimalNumber, DecimalNumber) メソッド [73 ページ]
2 番目の DecimalNumber オブジェクトを最初の DecimalNumber オブジェクトから減算し、その差を返します。

1.9.1 add(DecimalNumber, DecimalNumber) メソッド

2 つの DecimalNumber オブジェクトを加算し、その和を返します。

構文

```
public DecimalNumber add (
    DecimalNumber num1,
    DecimalNumber num2
) throws ULjException
```

パラメータ

num1 1 つの数値。
num2 別の数値。

戻り値

num1 と num2 の和。

1.9.2 divide(DecimalNumber, DecimalNumber) メソッド

最初の DecimalNumber オブジェクトを 2 番目の DecimalNumber オブジェクトで割って、その商を返します。

構文

```
public DecimalNumber divide (
    DecimalNumber num1,
    DecimalNumber num2
) throws ULjException
```

パラメータ

num1 被除数。
num2 除数。

戻り値

num1 を num2 で割った商。

1.9.3 getString() メソッド

DecimalNumber オブジェクトの String 表現を返します。

構文

```
public String getString () throws ULjException
```

戻り値

String 値。

1.9.4 isNull() メソッド

DecimalNumber オブジェクトが NULL かどうかを確認します。

構文

```
public boolean isNull ()
```

戻り値

オブジェクトが NULL の場合は true、NULL 以外の場合は false。

1.9.5 multiply(DecimalNumber, DecimalNumber) メソッド

2 つの DecimalNumber オブジェクトを乗算し、その積を返します。

構文

```
public DecimalNumber multiply (
    DecimalNumber num1,
    DecimalNumber num2
) throws ULjException
```

パラメータ

num1 被乗数。
num2 乗数。

戻り値

num1 と num2 の積。

1.9.6 set(String) メソッド

DecimalNumber オブジェクトに String 値を設定します。

構文

```
public void set (String value) throws ULjException
```

パラメータ

value String として表した数値。

1.9.7 setNull() メソッド

DecimalNumber オブジェクトを NULL に設定します。

構文

```
public void setNull () throws ULjException
```

1.9.8 subtract(DecimalNumber, DecimalNumber) メソッド

2 番目の DecimalNumber オブジェクトを最初の DecimalNumber オブジェクトから減算し、その差を返します。

構文

```
public DecimalNumber subtract (
    DecimalNumber num1,
    DecimalNumber num2
) throws ULjException
```

パラメータ

num1 被減数。

num2 減数。

戻り値

num1 と num2 の差。

1.10 Domain インタフェース

テーブル内のカラムの Domain オブジェクトの型情報を表します。

構文

```
public interface Domain
```

メンバー

Domain のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

変数

変更子とタイプ	変数	説明
public static final short	SHORT	16 ビット整数 (SMALLINT SQL 型) のドメイン ID 定数です。
public static final short	INTEGER	32 ビット整数 (INTEGER SQL 型) のドメイン ID 定数です。
public static final short	NUMERIC	合計桁数が固定 precision (サイズ)、小数点以下の桁数が scale 桁の数値 (NUMERIC(precision,scale) SQL 型) のドメイン ID 定数です。
public static final short	REAL	4 バイトの浮動小数点数 (REAL SQL 型) のドメイン ID 定数です。
public static final short	DOUBLE	8 バイトの浮動小数点数 (DOUBLE SQL 型) のドメイン ID 定数です。
public static final short	DATE	日付 (DATE SQL 型) のドメイン ID 定数です。
public static final short	VARCHAR	最大 size バイトの可変長文字オブジェクト (VARCHAR (size) SQL 型) のドメイン ID 定数です。
public static final short	LONGVARCHAR	任意の長いブロックの文字データ (CLOB) (LONG VARCHAR SQL 型) のドメイン ID 定数です。

変更子とタイプ	変数	説明
public static final short	BINARY	最大 size バイトの可変長のバイナリオブジェクト (BINARY (size) SQL 型) のドメイン ID 定数です。
public static final short	LONGBINARY	任意の長いブロックのバイナリデータ (BLOB) (LONG BINARY SQL 型) のドメイン ID 定数です。
public static final short	TIMESTAMP	タイムスタンプ (TIMESTAMP SQL 型) のドメイン ID 定数です。
public static final short	TIME	時刻 (TIME SQL 型) のドメイン ID 定数です。
public static final short	TINY	符号なし 8 ビット整数 (TINYINT SQL 型) のドメイン ID 定数です。
public static final short	BIG	64 ビット整数 (BIGINT SQL 型) のドメイン ID 定数です。
public static final short	UNSIGNED_INTEGER	符号なし 32 ビット整数 (UNSIGNED INTEGER SQL 型) のドメイン ID 定数です。
public static final short	UNSIGNED_SHORT	符号なし 16 ビット整数 (UNSIGNED SMALLINT SQL 型) のドメイン ID 定数です。
public static final short	UNSIGNED_BIG	符号なし 64 ビット整数 (UNSIGNED BIGINT SQL 型) のドメイン ID 定数です。
public static final short	BIT	ビット (BIT SQL 型) のドメイン ID 定数です。 BIT カラムはデフォルトでは NULL 入力不可です。
public static final short	UUID	UniqueIdentifier (UNIQUEIDENTIFIER SQL 型) のドメイン ID 定数です。
public static final short	ST_GEOMETRY	ジオメトリ (GEOMETRY SQL 型) のドメイン ID 定数です。
public static final short	LONGBINARYFILE	任意のデータファイルのドメイン ID 定数です。
public static final short	TIMESTAMP_ZONE	タイムゾーン付きタイムスタンプ (DATETIMEOFFSET SQL 型) のドメイン ID 定数です。
public static final short	DOMAIN_MAX	Domain 型の最大数です。

備考

このインタフェースには、さまざまなドメインを表す定数と、Domain オブジェクトから情報を抽出するためのメソッドがあります。

単純なデータベースのスキーマの作成例については、Connection インタフェースを参照してください。

型は次のように分類できます。

整数型:

ドメイン定数	SQL 型	値の範囲
BIT	BIT	0 または 1
TINY	TINYINT	0 ～ 255 (1 バイトの記憶領域を使用する符号なし整数)
SHORT	SMALLINT	-32768 ～ 32767 (2 バイトの記憶領域を使用する符号付き整数)
UNSIGNED_SHORT	UNSIGNED SMALLINT	0 ～ 65535 (2 バイトの記憶領域を使用する符号なし整数)
INTEGER	INTEGER	-2^{31} to 2^{31} - 1、または -2147483648 ～ 2147483647 (4 バイトの記憶領域を使用する符号付き整数)
UNSIGNED_INTEGER	UNSIGNED INTEGER	0 ～ 2^{32} - 1、または 0 ～ 4294967295 (4 バイトの記憶領域を使用する符号なし整数)
BIG	BIGINT	-2^{63} to 2^{63} - 1、または -9223372036854775808 ～ 9223372036854775807 (8 バイトの記憶領域を使用する符号付き整数)
UNSIGNED_BIG	UNSIGNED BIGINT	0 ～ 2^{64} - 1、または 0 ～ 18446744073709551615 (8 バイトの記憶領域を使用する符号なし整数)

整数以外の数値型:

ドメイン定数	SQL 型	値の範囲
REAL	REAL	-3.402823e+38 ～ 3.402823e+38、0 に最も近い最小の数値は 1.175495e-38 (4 バイトの記憶領域を使用する単精度の浮動小数点数、6 桁目の後に丸め誤差が生じる可能性があります)
DOUBLE	DOUBLE	-1.79769313486231e+308 ～ 1.79769313486231e+308、0 に最も近い最小の数値は 2.22507385850721e-308 (8 バイトの記憶領域を使用する単精度の浮動小数点数、15 桁目の後に丸め誤差が生じる可能性があります)
NUMERIC	NUMERIC(precision,scale)	合計桁数が *precision* (サイズ)、小数点以下の桁数が *scale* 桁の任意の 10 進数 (precision 内の丸めなし)

文字型とバイナリ型:

ドメイン定数	SQL 型	サイズの範囲
VARCHAR	VARCHAR(size)	1 ～ 32767 バイト (文字は 1 ～ 3 バイトの UTF-8 文字として格納)。式を評価するときのテンポラリ文字値の最大長は 2048 バイトです。
LONGVARCHAR	LONG VARCHAR	任意の長さ (メモリで許容される範囲内)。LONG VARCHAR カラムで実行可能な演算は、これらの挿入、更新、削除、またはクエリの select リストへのこれらの指定のみです。
BINARY	BINARY(size)	1 ～ 32767 バイト。式を評価するときのテンポラリ文字値の最大長は 2048 バイトです。
LONGBINARY	LONG BINARY	任意の長さ (メモリで許容される範囲内)。LONG BINARY カラムで実行可能な演算は、これらの挿入、更新、削除、またはクエリの select リストへのこれらの指定のみです。
UUID	UNIQUEIDENTIFIER	常に 16 バイトの解釈が特殊なバイナリ

日付型と時間型:

ドメイン定数	SQL 型	値
DATE	DATE	年、月、日。
TIME	TIME	時、分、秒 (小数位あり) で構成される時刻。
TIMESTAMP	TIMESTAMP	DATE と TIME。
TIMESTAMP_ZONE	TIMESTAMP_ZONE	タイムゾーン付きの DATE と TIME。

BIT カラムはデフォルトでは NULL 入力不可です。その他の型はデフォルトで NULL 入力可です。

関連情報

Connection インタフェース [31 ページ]

1.11 FileTransfer インタフェース

クライアントと Mobile Link サーバ間のファイル転送メカニズムを提供します。

構文

```
public interface FileTransfer
```

メンバー

FileTransfer のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public abstract boolean	downloadFile [81 ページ]	このオブジェクトの指定されたプロパティのファイルをダウンロードします。
public abstract String	getAuthenticationParms() [83 ページ]	カスタムのユーザ認証スクリプトに渡されるパラメータを返します。
public abstract int	getAuthStatus() [84 ページ]	前回行われたファイル転送の認証ステータスコードを返します。
public abstract long	getAuthValue() [84 ページ]	カスタムユーザ認証同期スクリプトで指定されている値を返します。
public abstract int	getFileAuthCode() [84 ページ]	前回行われたファイル転送の authenticate_file_transfer スクリプトからの戻り値を返します。
public abstract int	getLivenessTimeout() [85 ページ]	活性タイムアウトの長さを秒単位で返します。
public abstract String	getLocalFileName() [85 ページ]	ローカルファイル名を決定します。
public abstract String	getLocalPath() [86 ページ]	ローカルファイルシステムにおけるファイルの検索場所または格納場所を指定します。
public abstract String	getPassword() [86 ページ]	setUserName メソッドで指定されたユーザの Mobile Link パスワードを返します。
public abstract String	getRemoteKey() [87 ページ]	現在のリモートキーの値を決定します。
public abstract String	getServerFileName() [87 ページ]	サーバ上のファイルの名前を返します。
public abstract int	getStreamErrorCode() [88 ページ]	ストリームによってレポートされたエラーコードを返します。
public abstract String	getStreamErrorMessage() [89 ページ]	ストリーム自体によってレポートされるエラーメッセージを返します。
public abstract StreamHTTPParms	getStreamParms() [89 ページ]	同期ストリームの設定に使用するパラメータを返します。
public abstract String	getUserName() [90 ページ]	Mobile Link サーバがクライアントをユニークに識別する Mobile Link ユーザ名を返します。
public abstract String	getVersion() [90 ページ]	使用する同期スクリプトを返します。
public abstract boolean	isResumePartialTransfer() [91 ページ]	前の部分的な転送を再開するか、破棄するかを決定します。
public abstract boolean	isTransferredFile() [91 ページ]	前回行われたファイル転送時にファイルが実際にダウンロードされたかどうかを確認します。

変更子とタイプ	メソッド	説明
public abstract void	setAuthenticationParms(String) [92 ページ]	カスタムユーザ認証スクリプト (Mobile Link authenticate_parameters 接続イベント) のパラメータを指定します。
public abstract void	setLivenessTimeout(int) [92 ページ]	活性タイムアウトの長さを秒単位で設定します。
public abstract void	setLocalFileName(String) [93 ページ]	ファイルのローカル名を指定します。
public abstract void	setLocalPath(String) [94 ページ]	ローカルファイルシステムにおけるファイルの検索場所または格納場所を指定します。
public abstract void	setPassword(String) [95 ページ]	setUserName メソッドで指定されたユーザの Mobile Link パスワードを設定します。
public abstract void	setRemoteKey(String) [95 ページ]	リモートキーを指定します。
public abstract void	setResumePartialTransfer(boolean) [96 ページ]	前の部分的な転送を再開するか、破棄するかを指定します。
public abstract void	setServerFileName(String) [97 ページ]	サーバ上のファイルの名前を指定します。
public abstract void	setUserName(String) [98 ページ]	Mobile Link サーバがクライアントをユニークに識別する Mobile Link ユーザ名を設定します。
public abstract void	setVersion(String) [98 ページ]	使用する同期スクリプトを設定します。
public abstract boolean	uploadFile [99 ページ]	このオブジェクトの指定されたプロパティのファイルをアップロードします。

備考

FileTransfer オブジェクトは、DatabaseManager.createFileTransfer メソッドを呼び出すことによって取得します。

createFileTransfer メソッドによって返されたインスタンスを使用して、Mobile Link とローカルファイルシステム間で任意のファイルを転送できます。

ローカルファイルシステムは、メディアカードか、またはアプリケーションに適切なパーミッションがある内部ファイルシステム (/sdcard/Android/data/your.package.name/files/など)。

> **注記**
>
> アプリケーションで、同じローカルファイルに対して、2 つのダウンロードを同時に行うことはできません。

このセクションの内容:

downloadFile メソッド [81 ページ]
　このオブジェクトの指定されたプロパティのファイルをダウンロードします。

getAuthenticationParms() メソッド [83 ページ]
　カスタムのユーザ認証スクリプトに渡されるパラメータを返します。

getAuthStatus() メソッド [84 ページ]

前回行われたファイル転送の認証ステータスコードを返します。

getAuthValue() メソッド [84 ページ]
カスタムユーザ認証同期スクリプトで指定されている値を返します。

getFileAuthCode() メソッド [84 ページ]
前回行われたファイル転送の authenticate_file_transfer スクリプトからの戻り値を返します。

getLivenessTimeout() メソッド [85 ページ]
活性タイムアウトの長さを秒単位で返します。

getLocalFileName() メソッド [85 ページ]
ローカルファイル名を決定します。

getLocalPath() メソッド [86 ページ]
ローカルファイルシステムにおけるファイルの検索場所または格納場所を指定します。

getPassword() メソッド [86 ページ]
setUserName メソッドで指定されたユーザの Mobile Link パスワードを返します。

getRemoteKey() メソッド [87 ページ]
現在のリモートキーの値を決定します。

getServerFileName() メソッド [87 ページ]
サーバ上のファイルの名前を返します。

getStreamErrorCode() メソッド [88 ページ]
ストリームによってレポートされたエラーコードを返します。

getStreamErrorMessage() メソッド [89 ページ]
ストリーム自体によってレポートされるエラーメッセージを返します。

getStreamParms() メソッド [89 ページ]
同期ストリームの設定に使用するパラメータを返します。

getUserName() メソッド [90 ページ]
Mobile Link サーバがクライアントをユニークに識別する Mobile Link ユーザ名を返します。

getVersion() メソッド [90 ページ]
使用する同期スクリプトを返します。

isResumePartialTransfer() メソッド [91 ページ]
前の部分的な転送を再開するか、破棄するかを決定します。

isTransferredFile() メソッド [91 ページ]
前回行われたファイル転送時にファイルが実際にダウンロードされたかどうかを確認します。

setAuthenticationParms(String) メソッド [92 ページ]
カスタムユーザ認証スクリプト (Mobile Link authenticate_parameters 接続イベント) のパラメータを指定します。

setLivenessTimeout(int) メソッド [92 ページ]
活性タイムアウトの長さを秒単位で設定します。

setLocalFileName(String) メソッド [93 ページ]
ファイルのローカル名を指定します。

setLocalPath(String) メソッド [94 ページ]
ローカルファイルシステムにおけるファイルの検索場所または格納場所を指定します。

setPassword(String) メソッド [95 ページ]

setUserName メソッドで指定されたユーザの Mobile Link パスワードを設定します。

setRemoteKey(String) メソッド [95 ページ]
リモートキーを指定します。

setResumePartialTransfer(boolean) メソッド [96 ページ]
前の部分的な転送を再開するか、破棄するかを指定します。

setServerFileName(String) メソッド [97 ページ]
サーバ上のファイルの名前を指定します。

setUserName(String) メソッド [98 ページ]
Mobile Link サーバがクライアントをユニークに識別する Mobile Link ユーザ名を設定します。

setVersion(String) メソッド [98 ページ]
使用する同期スクリプトを設定します。

uploadFile メソッド [99 ページ]
このオブジェクトの指定されたプロパティのファイルをアップロードします。

関連情報

createFileTransfer(String, int, String, String) メソッド [66 ページ]

1.11.1 downloadFile メソッド

このオブジェクトの指定されたプロパティのファイルをダウンロードします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public abstract boolean	downloadFile() [82 ページ]	このオブジェクトの指定されたプロパティのファイルをダウンロードします。
public abstract boolean	downloadFile(FileTransferProgressListener) [82 ページ]	指定されたリスナに送信されるプログレスイベントとともに、このオブジェクトのプロパティで指定されたファイルをダウンロードします。

このセクションの内容:

downloadFile() メソッド [82 ページ]
このオブジェクトの指定されたプロパティのファイルをダウンロードします。

downloadFile(FileTransferProgressListener) メソッド [82 ページ]
指定されたリスナに送信されるプログレスイベントとともに、このオブジェクトのプロパティで指定されたファイルをダウンロードします。

1.11.1.1 downloadFile() メソッド

このオブジェクトの指定されたプロパティのファイルをダウンロードします。

構文

```
public abstract boolean downloadFile () throws ULjException
```

戻り値

ダウンロードに成功した場合は true、そうでない場合は ULjException がスローされメソッドは正常に返されません。

備考

setServerFileName メソッドで指定されたファイルは、指定のストリーム、userName、パスワード、スクリプトバージョンを使用して、setLocalPath メソッドで指定されたパスに、Mobile Link サーバからダウンロードされます。

setLocalFileName()、setAuthenticationParms()、setResumePartialTransfer() の各メソッドでは、他のオプションも指定できます。

getAuthStatus()、getAuthValue()、getFileAuthCode()、isTransferredFile()、getStreamErrorCode()、getStreamErrorMessage() の各メソッドを使用して、結果の詳細なステータスを取得できます。

1.11.1.2 downloadFile(FileTransferProgressListener) メソッド

指定されたリスナに送信されるプログレスイベントとともに、このオブジェクトのプロパティで指定されたファイルをダウンロードします。

構文

```
public abstract boolean downloadFile (FileTransferProgressListener listener)
throws ULjException
```

パラメータ

listener ファイル転送プログレスイベントを受信するオブジェクト。

戻り値

ダウンロードに成功した場合は true、そうでない場合は ULjException がスローされメソッドは正常に返されません。

備考

エラーが発生した場合、リスナにデータが送信されないことがあります。

関連情報

downloadFile() メソッド [82 ページ]

1.11.2 getAuthenticationParms() メソッド

カスタムのユーザ認証スクリプトに渡されるパラメータを返します。

構文

```
public abstract String getAuthenticationParms ()
```

戻り値

認証パラメータのリスト、またはパラメータが指定されていない場合は NULL。

関連情報

setAuthenticationParms(String) メソッド [92 ページ]

1.11.3 getAuthStatus() メソッド

前回行われたファイル転送の認証ステータスコードを返します。

構文

```
public abstract int getAuthStatus ()
```

戻り値

AuthStatusCode クラスの値。

1.11.4 getAuthValue() メソッド

カスタムユーザ認証同期スクリプトで指定されている値を返します。

構文

```
public abstract long getAuthValue ()
```

戻り値

カスタムユーザ認証同期スクリプトから返された整数。

1.11.5 getFileAuthCode() メソッド

前回行われたファイル転送の authenticate_file_transfer スクリプトからの戻り値を返します。

構文

```
public abstract int getFileAuthCode ()
```

戻り値

前回行われたファイル転送の authenticate_file_transfer スクリプトから返された整数。

1.11.6 getLivenessTimeout() メソッド

活性タイムアウトの長さを秒単位で返します。

構文

```
public abstract int getLivenessTimeout ()
```

戻り値

タイムアウト。

関連情報

setLivenessTimeout(int) メソッド [92 ページ]

1.11.7 getLocalFileName() メソッド

ローカルファイル名を決定します。

構文

```
public abstract String getLocalFileName ()
```

戻り値

ダウンロードしたファイルのローカルファイル名。

備考

ファイルのダウンロードの場合は、ダウンロードしたファイルの名前になります。ファイルのアップロードの場合は、アップロードするファイルの名前になります。

関連情報

setLocalFileName(String) メソッド [93 ページ]

1.11.8 getLocalPath() メソッド

ローカルファイルシステムにおけるファイルの検索場所または格納場所を指定します。

構文

```
public abstract String getLocalPath ()
```

戻り値

ローカルディレクトリ。

関連情報

setLocalPath(String) メソッド [94 ページ]

1.11.9 getPassword() メソッド

setUserName メソッドで指定されたユーザの Mobile Link パスワードを返します。

構文

```
public abstract String getPassword ()
```

戻り値

Mobile Link ユーザのパスワード。

関連情報

setPassword(String) メソッド [95 ページ]

1.11.10 getRemoteKey() メソッド

現在のリモートキーの値を決定します。

構文

```
public abstract String getRemoteKey ()
```

戻り値

リモートキーの値、またはリモートキーが指定されていない場合は NULL。

関連情報

setRemoteKey(String) メソッド [95 ページ]

1.11.11 getServerFileName() メソッド

サーバ上のファイルの名前を返します。

構文

```
public abstract String getServerFileName ()
```

戻り値

サーバ側のファイルの名前。

備考

ファイルのダウンロードの場合は、ダウンロードしたファイルの名前になります。ファイルのアップロードの場合は、アップロードするファイルの名前になります。

関連情報

setServerFileName(String) メソッド [97 ページ]

1.11.12 getStreamErrorCode() メソッド

ストリームによってレポートされたエラーコードを返します。

構文

```
public abstract int getStreamErrorCode ()
```

戻り値

通信ストリームエラーがなかった場合は 0、それ以外の場合はサーバからの応答コードを返します。

備考

エラーコードは HTTP の応答コードです。

1.11.13 getStreamErrorMessage() メソッド

ストリーム自体によってレポートされるエラーメッセージを返します。

構文

```
public abstract String getStreamErrorMessage ()
```

戻り値

メッセージがない場合は NULL、それ以外の場合は応答メッセージを返します。

備考

これは HTTP 応答メッセージです。

1.11.14 getStreamParms() メソッド

同期ストリームの設定に使用するパラメータを返します。

構文

```
public abstract StreamHTTPParms getStreamParms ()
```

戻り値

同期ストリームのパラメータを指定する StreamTCPIPParms オブジェクト。

備考

同期ストリームのタイプは、FileTransfer オブジェクトの作成時に指定します。

1.11.15 getUserName() メソッド

Mobile Link サーバがクライアントをユニークに識別する Mobile Link ユーザ名を返します。

構文

```
public abstract String getUserName ()
```

戻り値

Mobile Link ユーザ名

関連情報

setUserName(String) メソッド [98 ページ]

1.11.16 getVersion() メソッド

使用する同期スクリプトを返します。

構文

```
public abstract String getVersion ()
```

戻り値

スクリプトバージョン。

関連情報

setVersion(String) メソッド [98 ページ]

1.11.17 isResumePartialTransfer() メソッド

前の部分的な転送を再開するか、破棄するかを決定します。

構文

```
public abstract boolean isResumePartialTransfer ()
```

戻り値

ダウンロードを再開する場合は true、それ以外の場合は false。

関連情報

setResumePartialTransfer(boolean) メソッド [96 ページ]

1.11.18 isTransferredFile() メソッド

前回行われたファイル転送時にファイルが実際にダウンロードされたかどうかを確認します。

構文

```
public abstract boolean isTransferredFile ()
```

戻り値

ファイルが転送された場合は true、それ以外の場合は false。

備考

transferFile() メソッドが呼び出された時点でファイルがすでに最新だった場合、isTransferredFile() メソッドでは false が返されますが、このメソッドでは true が返されます。

エラーが発生し、transferFile() メソッドによって例外がスローされる場合は、isTransferredFile() メソッドにより false が返されます。

1.11.19 setAuthenticationParms(String) メソッド

カスタムユーザ認証スクリプト (Mobile Link authenticate_parameters 接続イベント) のパラメータを指定します。

構文

```
public abstract void setAuthenticationParms (String authParms) throws ULjException
```

パラメータ

authParms 認証パラメータのカンマ区切りのリスト、または NULL を参照してください。カンマ区切りのリストの詳細については、SyncParms クラスの説明を参照してください。

備考

最初の 255 文字列のみが使用されます。また、各文字列は 128 文字以下である必要があります (長すぎる文字列は、Mobile Link に送信されるときにトランケートされます)。

関連情報

getAuthenticationParms() メソッド [83 ページ]

1.11.20 setLivenessTimeout(int) メソッド

活性タイムアウトの長さを秒単位で設定します。

構文

```
public abstract void setLivenessTimeout (int timeout) throws ULjException
```

パラメータ

timeout 新しい活性タイムアウト値。

備考

活性タイムアウトは、サーバで許容される、リモートのアイドル時間の長さです。リモートが 1 秒間サーバと通信しなかった場合、サーバはリモートとの接続が失われたとみなし、ファイル転送を終了します。リモートは、接続を継続するために、自動的に定期メッセージをサーバに送信します。

負の値を設定すると、例外がスローされます。値は Mobile Link サーバによって予告なく変更される場合があります。変更は、値が低すぎるか高すぎる場合に行われます。

デフォルト値は 240 秒です。

関連情報

getLivenessTimeout() メソッド [85 ページ]

1.11.21 setLocalFileName(String) メソッド

ファイルのローカル名を指定します。

構文

```
public abstract void setLocalFileName (String localFileName)
```

パラメータ

localFileName ダウンロードファイルのローカル名を指定する文字列。値が NULL 参照の場合は、fileName が使用されます。デフォルトは NULL 参照です。

備考

ファイルのダウンロードの場合は、ダウンロードしたファイルの名前になります。ファイルのアップロードの場合は、アップロードするファイルの名前になります。ファイル名にドライブまたはパスの情報を含めることはできません。

関連情報

getLocalFileName() メソッド [85 ページ]

setLocalPath(String) メソッド [94 ページ]

1.11.22 setLocalPath(String) メソッド

ローカルファイルシステムにおけるファイルの検索場所または格納場所を指定します。

構文

```
public abstract void setLocalPath (String localPath)
```

パラメータ

localPath ファイルのローカルディレクトリを指定する文字列。デフォルトは NULL 参照です。

備考

ローカルディレクトリの構文は、プラットフォームによって異なります。

- デスクトップの場合の構文は、"C:¥¥ulj¥¥" のようになります。
- Android ファイルシステムの場合の構文は、"/sdcard/Android/data/your.package.name/files/" のようになります。

デフォルトのローカルディレクトリも、デバイスのオペレーティングシステムによって異なります。

- デスクトップコンピュータでは、localPath パラメータが NULL の場合、ファイルは現在のディレクトリに格納されます。
- Android ファイルシステムストアの場合、localPath パラメータにデフォルトの値がないので、明示的に設定する必要があります。

関連情報

getLocalPath() メソッド [86 ページ]
setLocalFileName(String) メソッド [93 ページ]

1.11.23 setPassword(String) メソッド

setUserName メソッドで指定されたユーザの Mobile Link パスワードを設定します。

構文

```
public abstract void setPassword (String password) throws ULjException
```

パラメータ

password Mobile Link ユーザのパスワード。

備考

このユーザ名とパスワードは、データベースのユーザ ID とパスワードとは異なります。このメソッドは、Mobile Link サーバに対してアプリケーションを認証するために使用されます。

デフォルトは空の文字列で、これはパスワードなしを示します。

関連情報

getPassword() メソッド [86 ページ]
setUserName(String) メソッド [98 ページ]

1.11.24 setRemoteKey(String) メソッド

リモートキーを指定します。

構文

```
public abstract void setRemoteKey (String remoteKey)
```

パラメータ

remoteKey リモートキーの値、またはリモートキーが指定されていない場合は NULL。

備考

リモートキーは、サーバの authenticate_file_upload スクリプトに渡されるパラメータです。

スクリプトでは、このパラメータを使用して、サーバに格納するファイルの名前と場所を決定します。

この値が指定されていない場合は、getUserName() の値がリモートキーとして使用されます。

関連情報

getRemoteKey() メソッド [87 ページ]

1.11.25 setResumePartialTransfer(boolean) メソッド

前の部分的な転送を再開するか、破棄するかを指定します。

構文

```
public abstract void setResumePartialTransfer (boolean resume)
```

パラメータ

resume 前回の部分的なダウンロードを再開する場合は true、破棄する場合は false に設定します。

備考

デフォルトは true です。

Ultra Light J では、通信エラーや、ユーザによる FileTransferProgressListener オブジェクトからのアボートが原因で失敗したファイル転送を再起動できます。

ファイルのダウンロードの場合、Ultra Light は、ダウンロードを受信しながら処理します。ダウンロードが中断した場合は、部分的なダウンロードファイルが保持されるため、次の転送中に再開できます。ファイルがサーバ上で更新されている場合は、部分的なダウンロードは破棄され、新しくダウンロードが開始されます。

ファイルのアップロードの場合、以降のファイルのアップロードで前回のアップロードを再開できるようにするため、Mobile Link サーバは部分的にアップロードされたファイルを保持します。ただし、ファイルがローカルで更新されている場合は、部分的なアップロードは破棄され、新しくアップロードが開始されます。

関連情報

isResumePartialTransfer() メソッド [91 ページ]

1.11.26 setServerFileName(String) メソッド

サーバ上のファイルの名前を指定します。

構文

```
public abstract void setServerFileName (String fileName) throws ULjException
```

パラメータ

fileName Mobile Link サーバによって認識されたファイルの名前を指定する文字列。

備考

ファイルのダウンロードの場合は、ダウンロードしたファイルの名前になります。ファイルのアップロードの場合は、アップロードするファイルの名前になります。

このパラメータは、FileTransfer オブジェクトの作成時に初期化されます。

Mobile Link は、指定されたファイルを最初に userName サブディレクトリで、次にルートディレクトリで検索します。ルートダウンロードディレクトリは、Mobile Link サーバの -ftr オプションで指定され、ルートアップロードディレクトリは -ftru オプションで指定されます。

fileName にはドライブまたはパスの情報を含めないでください。そのような情報を含めると、ファイルが見つからなくなります。たとえば、"myfile.txt" は有効ですが、"somedir¥myfile.txt"、"..¥myfile.txt"、"c:¥myfile.txt" は無効です。

関連情報

getServerFileName() メソッド [87 ページ]

1.11.27 setUserName(String) メソッド

Mobile Link サーバがクライアントをユニークに識別する Mobile Link ユーザ名を設定します。

構文

```
public abstract void setUserName (String userName) throws ULjException
```

パラメータ

userName Mobile Link ユーザ名

備考

Mobile Link サーバは、この値を使用して、サーバ側でファイルを特定します。Mobile Link ユーザ名とパスワードは他のデータベースユーザ ID やパスワードとは別のもので、アプリケーションを Mobile Link サーバに対して識別し、認証するために使用されます。

このパラメータは、FileTransfer オブジェクトの作成時に初期化されます。

関連情報

getUserName() メソッド [90 ページ]
setPassword(String) メソッド [95 ページ]

1.11.28 setVersion(String) メソッド

使用する同期スクリプトを設定します。

構文

```
public abstract void setVersion (String version) throws ULjException
```

パラメータ

version スクリプトバージョン。

備考

統合データベースの同期スクリプトは、それぞれバージョン文字列で区別されます。Ultra Light J アプリケーションは、バージョン文字列により、同期スクリプトのセットから選択できます。

このパラメータは、FileTransfer オブジェクトの作成時に初期化されます。

関連情報

getVersion() メソッド [90 ページ]

1.11.29 uploadFile メソッド

このオブジェクトの指定されたプロパティのファイルをアップロードします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public abstract boolean	uploadFile() [100 ページ]	このオブジェクトの指定されたプロパティのファイルをアップロードします。
public abstract boolean	uploadFile(FileTransferProgressListener) [100 ページ]	指定されたリスナに送信されるプログレスイベントとともに、このオブジェクトのプロパティで指定されたファイルをアップロードします。

このセクションの内容:

1.11.29.1 uploadFile() メソッド

このオブジェクトの指定されたプロパティのファイルをアップロードします。

構文

```
public abstract boolean uploadFile () throws ULjException
```

戻り値

アップロードに成功した場合は true、そうでない場合は ULjException がスローされメソッドは正常に返されません。

備考

setLocalFileName メソッドと setLocalPath メソッドで指定されたファイルは、指定のストリーム、ユーザ名、パスワード、スクリプトバージョンを使用して、Mobile Link サーバの、setServerFileName メソッドで指定されたファイルにアップロードされます。

setAuthenticationParms() メソッドと setResumePartialTransfer() メソッドでは、他のオプションも指定できます。

getAuthStatus()、getAuthValue()、getFileAuthCode()、isTransferredFile()、getStreamErrorCode()、getStreamErrorMessage() の各メソッドを使用して、結果の詳細なステータスを取得できます。

1.11.29.2 uploadFile(FileTransferProgressListener) メソッド

指定されたリスナに送信されるプログレスイベントとともに、このオブジェクトのプロパティで指定されたファイルをアップロードします。

構文

```
public abstract boolean uploadFile (FileTransferProgressListener listener) throws
ULjException
```

パラメータ

listener ファイル転送プログレスイベントを受信するオブジェクト。

戻り値

アップロードに成功した場合は true、そうでない場合は ULjException がスローされメソッドは正常に返されません。

備考

エラーが発生した場合、リスナにデータが送信されないことがあります。

関連情報

uploadFile() メソッド [100 ページ]

1.12 FileTransferProgressData インタフェース

ファイル転送の進行状況のモニタリングデータを通知します。

構文

```
public interface FileTransferProgressData
```

メンバー

FileTransferProgressData のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public abstract long	getBytesTransferred() [102 ページ]	現在までに転送されたバイト数を返します。
public abstract long	getFileSize() [102 ページ]	転送されるファイルのサイズを返します。
public abstract long	getResumedAtSize() [103 ページ]	転送が再開されるファイル内のポイントを返します。

このセクションの内容:

転送されるファイルのサイズを返します。

getResumedAtSize() メソッド [103 ページ]
転送が再開されるファイル内のポイントを返します。

1.12.1 getBytesTransferred() メソッド

現在までに転送されたバイト数を返します。

構文

```
public abstract long getBytesTransferred ()
```

戻り値

現在までに転送されたバイト数。

備考

このメソッドでは、現在のファイル転送セッションによって転送されたバイト数のカウントに、以前に中断された転送によって転送されたバイト数が加算されます。

getResumedAtSize メソッドによって返された値を引くと、現在のセッションによって転送されたバイト数を算出できます。

関連情報

getResumedAtSize() メソッド [103 ページ]

1.12.2 getFileSize() メソッド

転送されるファイルのサイズを返します。

構文

```
public abstract long getFileSize ()
```

戻り値

ファイルのサイズ (バイト単位)。

備考

戻り値は、ファイル転送セッションの継続中は一定の値になります。

1.12.3 getResumedAtSize() メソッド

転送が再開されるファイル内のポイントを返します。

構文

```
public abstract long getResumedAtSize ()
```

戻り値

以前に転送されたバイト数。

備考

戻り値は、ファイル転送セッションの継続中は一定の値になります。

1.13 FileTransferProgressListener インタフェース

ファイル転送の進行状況イベントを受信します。

構文

```
public interface FileTransferProgressListener
```

メンバー

FileTransferProgressListener のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public boolean	fileTransferProgressed(FileTransferProgressData) [104 ページ]	ユーザに転送の進行状況を通知するために、ファイル転送中に呼び出されます。

備考

ファイル転送中に進行状況レポートを受信するために、新しいクラスが作成されます。

次の例は、FileTransferProgressListener インタフェースを実装する単純な SyncObserver インタフェースを示しています。

```
class MyObserver implements FileTransferProgressListener {
  public boolean fileTransferProgressed( FileTransferProgressData data ) {
      System.out.println(
          "file transfer progress "
          + " bytes received = " + data.getBytesTransferred()
      );
      return false;   // Always continue file transfer.
  }
  public MyObserver() {} // The default constructor.
}
```

このセクションの内容:

fileTransferProgressed(FileTransferProgressData) メソッド [104 ページ]
ユーザに転送の進行状況を通知するために、ファイル転送中に呼び出されます。

1.13.1 fileTransferProgressed(FileTransferProgressData) メソッド

ユーザに転送の進行状況を通知するために、ファイル転送中に呼び出されます。

構文

```
public boolean fileTransferProgressed (FileTransferProgressData data)
```

パラメータ

data 最新のファイル転送プログレスデータを保持している FileTransferProgressData オブジェクト。

戻り値

転送をキャンセルする場合は true を、続行する場合は false を返します。

備考

リスナは、次の状況で呼び出されます。

- 最初のディスク書き込みの前
- ディスク書き込みごとまたは 0.5 秒ごとのどちらか後のほう
- ファイルのダウンロードの完了後

通常、キャンセル要求は Ultra Light J API で受け入れられます。その結果、errorCode が ULjException.SQLE_INTERRUPTED 定数に設定された ULjException オブジェクトがスローされます。

fileTransferProgressed の呼び出し中に、Ultra Light J API のメソッドを呼び出さないでください。

1.14 IndexSchema インタフェース

インデックスのスキーマを指定し、システムテーブルの問い合わせに便利な定数を提供します。

構文

```
public interface IndexSchema
```

メンバー

IndexSchema のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

変数

変更子とタイプ	変数	説明
public static final byte	ASCENDING	カラムのインデックスが昇順でソートされます。
public static final byte	DESCENDING	カラムのインデックスが降順でソートされます。

変更子とタイプ	変数	説明
public static final byte	UNIQUE_KEY	インデックスがユニークキーであることを示します。 この値は、SYS_TABLES システムテーブルの table_flags カラムで他のフラグと論理的に組み合わせることができます。
public static final byte	UNIQUE_INDEX	インデックスがユニークインデックスであることを示します。 この値は、SYS_TABLES システムテーブルの table_flags カラムで他のフラグと論理的に組み合わせることができます。
public static final byte	PERSISTENT	インデックスが永続的であることを示します。 この値は、SYS_TABLES システムテーブルの table_flags カラムで他のフラグと論理的に組み合わせることができます。
public static final byte	PRIMARY_INDEX	インデックスがプライマリキーであることを示します。 この値は、SYS_TABLES システムテーブルの table_flags カラムで他のフラグと論理的に組み合わせることができます。

備考

このインタフェースには、インデックスのフラグやとソート順など、インデックス関連の定数のみが含まれます。

1.15 PreparedStatement インタフェース

SQL クエリを実行して ResultSet オブジェクトを生成するか、準備された SQL 文をデータベースに対して実行するメソッドを提供します。

構文

```
public interface PreparedStatement
```

メンバー

PreparedStatement のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public void	close() [109 ページ]	PreparedStatement を閉じて、関連付けられているメモリリソースを解放します。
public boolean	execute() [109 ページ]	準備された SQL 文を実行します。
public ResultSet	executeQuery() [110 ページ]	準備された SQL SELECT 文を実行し、ResultSet オブジェクトを返します。
public java.io.OutputStream	getBlobOutputStream [110 ページ]	OutputStream オブジェクトを返します。
public java.io.Writer	getClobWriter [112 ページ]	Writer オブジェクトを返します。
public int	getOrdinal(String) [113 ページ]	name で指定された値の (1 から始まる) 順序を返します。
public short	getParameterCount() [114 ページ]	この文の入力パラメータの数を取得します。
public short	getParameterType(int) [114 ページ]	パラメータのドメインの型を取得します。
public String	getPlan() [114 ページ]	SQL クエリ実行プランのテキストベースの記述を返します。
public String	getPlanTree() [115 ページ]	SQL クエリ実行プランのテキストベースの記述をツリー形式で返します。
public ResultSet	getResultSet() [116 ページ]	準備された SQL 文の ResultSet オブジェクトを返します。
public int	getUpdateCount() [117 ページ]	最後の実行文の後に挿入、更新、または削除された行数を返します。
public boolean	hasResultSet() [117 ページ]	PreparedStatement オブジェクトに ResultSet オブジェクトが含まれるかどうかを確認します。
public void	set [118 ページ]	SQL 文のホスト変数に値を設定します。
public void	setNull [129 ページ]	SQL 文内の name で定義されたホスト変数に NULL 値を設定します。

備考

次の例は、PreparedStatement オブジェクトを実行し、SELECT 文によって ResultSet オブジェクトが作成されたかどうかを確認し、ResultSet オブジェクトをローカル変数に格納し、PreparedStatement を閉じる方法を示しています。

```
// Create a new PreparedStatement object from an existing connection.
String sql_string = "SELECT * FROM SampleTable";
PreparedStatement ps = conn.prepareStatement(sql_string);
// Result returns true if the statement runs successfully.
boolean result = ps.execute();
// Check if the PreparedStatement object contains a ResultSet object.
if (ps.hasResultSet()) {
    // Store the ResultSet in the rs variable.
    ResultSet rs = ps.getResultSet;
}
// Close the PreparedStatement object to release resources.
ps.close();
```

文に式が含まれる場合、カラム名があるところにはどこでもホスト変数が含まれる可能性があります。ホスト変数は、*?* 文字 (名前なしホスト変数) または *:name* (名前付きホスト変数) のいずれかとして入力されます。

次の例には、対象の SQL 文用に準備された PreparedStatement オブジェクトを使って設定できる 2 つのホスト変数があります。

```
SELECT * FROM SampleTable WHERE pk > :bound AND pk < ?
```

このセクションの内容:

関連情報

Connection インタフェース [31 ページ]
prepareStatement(String) メソッド [50 ページ]

1.15.1 close() メソッド

PreparedStatement を閉じて、関連付けられているメモリリソースを解放します。

構文

```
public void close () throws ULjException
```

備考

このオブジェクトに対してこれ以上メソッドを使用できなくなります。PreparedStatement オブジェクトに ResultSet オブジェクトが含まれている場合は、両方のオブジェクトが閉じます。

1.15.2 execute() メソッド

準備された SQL 文を実行します。

構文

```
public boolean execute () throws ULjException
```

戻り値

文が正常に実行された場合は true、それ以外の場合は false。

1.15.3 executeQuery() メソッド

準備された SQL SELECT 文を実行し、ResultSet オブジェクトを返します。

構文

```
public ResultSet executeQuery () throws ULjException
```

戻り値

準備された SQL SELECT 文のクエリの結果を含む ResultSet オブジェクト。

関連情報

ResultSet インタフェース [130 ページ]

1.15.4 getBlobOutputStream メソッド

OutputStream オブジェクトを返します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public java.io.OutputStream	getBlobOutputStream(String) [111 ページ]	OutputStream オブジェクトを返します。
public java.io.OutputStream	getBlobOutputStream(int) [111 ページ]	OutputStream オブジェクトを返します。

このセクションの内容:

1.15.4.1 getBlobOutputStream(String) メソッド

OutputStream オブジェクトを返します。

構文

```
public java.io.OutputStream getBlobOutputStream (String name) throws ULjException
```

パラメータ

name ホスト変数名を表す文字列。

戻り値

指定された値の OutputStream オブジェクト。

1.15.4.2 getBlobOutputStream(int) メソッド

OutputStream オブジェクトを返します。

構文

```
public java.io.OutputStream getBlobOutputStream (int ordinal) throws ULjException
```

パラメータ

ordinal SQL 文でのホスト変数の順序を表す 1 から始まる整数。

戻り値

指定された値の OutputStream オブジェクト。

1.15.5 getClobWriter メソッド

Writer オブジェクトを返します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public java.io.Writer	getClobWriter(String) [112 ページ]	Writer オブジェクトを返します。
public java.io.Writer	getClobWriter(int) [113 ページ]	Writer オブジェクトを返します。

このセクションの内容:

getClobWriter(String) メソッド [112 ページ]
Writer オブジェクトを返します。

getClobWriter(int) メソッド [113 ページ]
Writer オブジェクトを返します。

1.15.5.1 getClobWriter(String) メソッド

Writer オブジェクトを返します。

構文

```
public java.io.Writer getClobWriter (String name) throws ULjException
```

パラメータ

name ホスト変数名を表す文字列。

戻り値

指定された値の Writer オブジェクト。

1.15.5.2 getClobWriter(int) メソッド

Writer オブジェクトを返します。

構文

```
public java.io.Writer getClobWriter (int ordinal) throws ULjException
```

パラメータ

ordinal SQL 文でのホスト変数の順序を表す 1 から始まる整数。

戻り値

指定された値の Writer オブジェクト。

1.15.6 getOrdinal(String) メソッド

name で指定された値の (1 から始まる) 順序を返します。

構文

```
public int getOrdinal (String name) throws ULjException
```

パラメータ

name テーブルのカラム名を表す文字列。

戻り値

name で指定された値の (1 から始まる) 順序。

1.15.7 getParameterCount() メソッド

準備した文の入力パラメータの数を取得します。

構文

```
public short getParameterCount () throws ULjException
```

戻り値

準備した文の入力パラメータの数。

1.15.8 getParameterType(int) メソッド

パラメータのドメインの型を取得します。

構文

```
public short getParameterType (int ordinal) throws ULjException
```

パラメータ

ordinal パラメータの、1 から始まる序数。

戻り値

指定したパラメータのドメインタイプ。

1.15.9 getPlan() メソッド

SQL クエリ実行プランのテキストベースの記述を返します。

構文

```
public String getPlan () throws ULjException
```

戻り値

プランの String 表現。

備考

このメソッドは、開発中の使用を目的とします。

このプランには、getPlanTree メソッドで示される情報と同じものが含まれます。違いは表示形式です。

プランがない場合は、空の文字列を返します。準備された文が SQL クエリの場合には、プランが存在します。

関連するクエリの実行前にプランが取得された場合は、クエリの実行に使用される操作がプランに表示されます。また、クエリの実行後にプランが取得された場合は、各操作で生成されるロー数も表示されます。このプランを使用して、クエリの実行に関する理解を深めることができます。

次に、String として表されたプランツリーの例を示します。構造を表す '|' 文字を使用して複数行に表示されます。

```
SELECT * FROM tab1, tab2 WHERE col1 > pk2
row: 2 20 10 banana
row: 3 30 10 banana
row: 4 40 10 banana
row: 4 40 30 peach
row: 5 50 10 banana
row: 5 50 30 peach
row: 5 50 40 apple
plan: root:7(inner-join:7(table-scan:5[tab1,prime_key],index-scan:
7[tab2,prime_key]))
```

関連情報

getPlanTree() メソッド [115 ページ]

1.15.10 getPlanTree() メソッド

SQL クエリ実行プランのテキストベースの記述をツリー形式で返します。

構文

```
public String getPlanTree () throws ULjException
```

戻り値

ツリーで表されるプランの String 表現。

備考

このメソッドは、開発中の使用を目的とします。

このプランには、getPlan メソッドで示される情報と同じものが含まれます。違いは表示形式です。

プランがない場合は、空の文字列を返します。準備された文が SQL クエリの場合には、プランが存在します。

関連するクエリの実行前にプランが取得された場合は、クエリの実行に使用される操作がプランに表示されます。また、クエリの実行後にプランが取得された場合は、各操作で生成されるロー数も表示されます。このプランを使用して、クエリの実行に関する理解を深めることができます。

次に、String として表されたプランツリーの例を示します。構造を表す '|' 文字を使用して複数行に表示されます。

```
SELECT * FROM tab1, tab2 WHERE col1 > pk2
row: 2 20 10 banana
row: 3 30 10 banana
row: 4 40 10 banana
row: 4 40 30 peach
row: 5 50 10 banana
row: 5 50 30 peach
row: 5 50 40 apple
plan:
root:7
|
inner-join:7
| |
| index-scan:7[tab2,prime_key]
|
table-scan:5[tab1,prime_key]
```

関連情報

getPlan() メソッド [114 ページ]

1.15.11 getResultSet() メソッド

準備された SQL 文の ResultSet オブジェクトを返します。

構文

```
public ResultSet getResultSet () throws ULjException
```

戻り値

準備された SQL 文のクエリの結果を含む ResultSet オブジェクト。

関連情報

ResultSet インタフェース [130 ページ]

1.15.12 getUpdateCount() メソッド

最後の実行文の後に挿入、更新、または削除されたロー数を返します。

構文

```
public int getUpdateCount () throws ULjException
```

戻り値

文で変更を実行できない場合は -1 を返し、そうでない場合は変更されたロー数を返します。

1.15.13 hasResultSet() メソッド

PreparedStatement オブジェクトに ResultSet オブジェクトが含まれるかどうかを確認します。

構文

```
public boolean hasResultSet () throws ULjException
```

戻り値

ResultSet が見つかった場合は true、それ以外の場合は false。

関連情報

ResultSet インタフェース [130 ページ]

1.15.14 set メソッド

SQL 文のホスト変数に値を設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public void	set(String, Date) [120 ページ]	SQL 文内の namel で定義されたホスト変数に java.util.Date を設定します。
public void	set(String, DecimalNumber) [121 ページ]	SQL 文内の name で定義されたホスト変数に DecimalNumber をオブジェクトを設定します。
public void	set(String, String) [121 ページ]	SQL 文内の name で定義されたホスト変数に String 値を設定します。
public void	set(String, UUIDValue) [121 ページ]	SQL 文内の name で定義されたホスト変数に UUIDValue 値を設定します。
public void	set(String, boolean) [122 ページ]	SQL 文内の name で定義されたホスト変数に boolean 値を設定します。
public void	set(String, byte[]) [122 ページ]	SQL 文内の name で定義されたホスト変数に byte 配列値を設定します。
public void	set(String, double) [123 ページ]	SQL 文内の name で定義されたホスト変数に double 値を設定します。
public void	set(String, float) [123 ページ]	SQL 文内の name で定義されたホスト変数に float 値を設定します。
public void	set(String, int) [124 ページ]	SQL 文内の name で定義されたホスト変数に int 値を設定します。
public void	set(String, long) [124 ページ]	SQL 文内の name で定義されたホスト変数に long 型の整数値を設定します。
public void	set(int, Date) [124 ページ]	SQL 文内の ordinal で定義されたホスト変数に java.util.Date を設定します。
public void	set(int, DecimalNumber) [125 ページ]	SQL 文内の ordinal で定義されたホスト変数に DecimalNumber オブジェクトを設定します。

変更子とタイプ	オーバーロード名	説明
public void	set(int, String) [125 ページ]	SQL 文内の ordinal で定義されたホスト変数に String 値を設定します。
public void	set(int, UUIDValue) [126 ページ]	SQL 文内の ordinal で定義されたホスト変数に UUIDValue 値を設定します。
public void	set(int, boolean) [126 ページ]	SQL 文内の ordinal で定義されたホスト変数に boolean 値を設定します。
public void	set(int, byte[]) [127 ページ]	SQL 文内の ordinal で定義されたホスト変数に byte 配列値を設定します。
public void	set(int, double) [127 ページ]	SQL 文内の ordinal で定義されたホスト変数に double 値を設定します。
public void	set(int, float) [127 ページ]	SQL 文内の ordinal で定義されたホスト変数に float 値を設定します。
public void	set(int, int) [128 ページ]	SQL 文内の ordinal で定義されたホスト変数に int 値を設定します。
public void	set(int, long) [128 ページ]	SQL 文内の ordinal で定義されたホスト変数に long 型の整数値を設定します。

このセクションの内容:

set(String, Date) メソッド [120 ページ]
SQL 文内の namel で定義されたホスト変数に java.util.Date を設定します。

set(String, DecimalNumber) メソッド [121 ページ]
SQL 文内の name で定義されたホスト変数に DecimalNumber をオブジェクトを設定します。

set(String, String) メソッド [121 ページ]
SQL 文内の name で定義されたホスト変数に String 値を設定します。

set(String, UUIDValue) メソッド [121 ページ]
SQL 文内の name で定義されたホスト変数に UUIDValue 値を設定します。

set(String, boolean) メソッド [122 ページ]
SQL 文内の name で定義されたホスト変数に boolean 値を設定します。

set(String, byte[]) メソッド [122 ページ]
SQL 文内の name で定義されたホスト変数に byte 配列値を設定します。

set(String, double) メソッド [123 ページ]
SQL 文内の name で定義されたホスト変数に double 値を設定します。

set(String, float) メソッド [123 ページ]
SQL 文内の name で定義されたホスト変数に float 値を設定します。

set(String, int) メソッド [124 ページ]
SQL 文内の name で定義されたホスト変数に int 値を設定します。

set(String, long) メソッド [124 ページ]
SQL 文内の name で定義されたホスト変数に long 型の整数値を設定します。

set(int, Date) メソッド [124 ページ]
SQL 文内の ordinal で定義されたホスト変数に java.util.Date を設定します。

set(int, DecimalNumber) メソッド [125 ページ]
SQL 文内の ordinal で定義されたホスト変数に DecimalNumber オブジェクトを設定します。

set(int, String) メソッド [125 ページ]
SQL 文内の ordinal で定義されたホスト変数に String 値を設定します。

set(int, UUIDValue) メソッド [126 ページ]
SQL 文内の ordinal で定義されたホスト変数に UUIDValue 値を設定します。

set(int, boolean) メソッド [126 ページ]
SQL 文内の ordinal で定義されたホスト変数に boolean 値を設定します。

set(int, byte[]) メソッド [127 ページ]
SQL 文内の ordinal で定義されたホスト変数に byte 配列値を設定します。

set(int, double) メソッド [127 ページ]
SQL 文内の ordinal で定義されたホスト変数に double 値を設定します。

set(int, float) メソッド [127 ページ]
SQL 文内の ordinal で定義されたホスト変数に float 値を設定します。

set(int, int) メソッド [128 ページ]
SQL 文内の ordinal で定義されたホスト変数に int 値を設定します。

set(int, long) メソッド [128 ページ]
SQL 文内の ordinal で定義されたホスト変数に long 型の整数値を設定します。

1.15.14.1 set(String, Date) メソッド

SQL 文内の namel で定義されたホスト変数に java.util.Date を設定します。

構文

```
public void set (
    String name,
    java.util.Date value
) throws ULjException
```

パラメータ

name ホスト変数名を表す文字列。
value 設定する値。

1.15.14.2 set(String, DecimalNumber) メソッド

SQL 文内の name で定義されたホスト変数に DecimalNumber をオブジェクトを設定します。

構文

```
public void set (
    String name,
    DecimalNumber value
) throws ULjException
```

パラメータ

name ホスト変数名を表す文字列。
value 設定する DecimalNumber の値。

1.15.14.3 set(String, String) メソッド

SQL 文内の name で定義されたホスト変数に String 値を設定します。

構文

```
public void set (
    String name,
    String value
) throws ULjException
```

パラメータ

name ホスト変数名を表す文字列。
value 設定する値。

1.15.14.4 set(String, UUIDValue) メソッド

SQL 文内の name で定義されたホスト変数に UUIDValue 値を設定します。

構文

```
public void set (
```

```
    String name,
    UUIDValue value
) throws ULjException
```

パラメータ

name ホスト変数名を表す文字列。
value 設定する値。

1.15.14.5 set(String, boolean) メソッド

SQL 文内の name で定義されたホスト変数に boolean 値を設定します。

構文

```
public void set (
    String name,
    boolean value
) throws ULjException
```

パラメータ

name ホスト変数名を表す文字列。
value 設定する値。

1.15.14.6 set(String, byte[]) メソッド

SQL 文内の name で定義されたホスト変数に byte 配列値を設定します。

構文

```
public void set (
    String name,
    byte[] value
) throws ULjException
```

パラメータ

name ホスト変数名を表す文字列。
value 設定する値。

1.15.14.7 set(String, double) メソッド

SQL 文内の name で定義されたホスト変数に double 値を設定します。

構文

```
public void set (
    String name,
    double value
) throws ULjException
```

パラメータ

name ホスト変数名を表す文字列。
value 設定する値。

1.15.14.8 set(String, float) メソッド

SQL 文内の name で定義されたホスト変数に float 値を設定します。

構文

```
public void set (
    String name,
    float value
) throws ULjException
```

パラメータ

name ホスト変数名を表す文字列。
value 設定する値。

1.15.14.9 set(String, int) メソッド

SQL 文内の name で定義されたホスト変数に int 値を設定します。

構文

```
public void set (
    String name,
    int value
) throws ULjException
```

パラメータ

name ホスト変数名を表す文字列。
value 設定する値。

1.15.14.10 set(String, long) メソッド

SQL 文内の name で定義されたホスト変数に long 型の整数値を設定します。

構文

```
public void set (
    String name,
    long value
) throws ULjException
```

パラメータ

name ホスト変数名を表す文字列。
value 設定する値。

1.15.14.11 set(int, Date) メソッド

SQL 文内の ordinal で定義されたホスト変数に java.util.Date を設定します。

構文

```
public void set (
```

```
    int ordinal,
    java.util.Date value
) throws ULjException
```

パラメータ

ordinal SQL 文でのホスト変数の順序を表す 1 から始まる整数。
value 設定する値。

1.15.14.12 set(int, DecimalNumber) メソッド

SQL 文内の ordinal で定義されたホスト変数に DecimalNumber オブジェクトを設定します。

構文

```
public void set (
    int ordinal,
    DecimalNumber value
) throws ULjException
```

パラメータ

ordinal SQL 文でのホスト変数の順序を表す 1 から始まる整数。
value 設定する DecimalNumber の値。

1.15.14.13 set(int, String) メソッド

SQL 文内の ordinal で定義されたホスト変数に String 値を設定します。

構文

```
public void set (
    int ordinal,
    String value
) throws ULjException
```

パラメータ

ordinal SQL 文でのホスト変数の順序を表す 1 から始まる整数。
value 設定する値。

1.15.14.14 set(int, UUIDValue) メソッド

SQL 文内の ordinal で定義されたホスト変数に UUIDValue 値を設定します。

構文

```
public void set (
    int ordinal,
    UUIDValue value
) throws ULjException
```

パラメータ

ordinal SQL 文でのホスト変数の順序を表す 1 から始まる整数。
value 設定する値。

1.15.14.15 set(int, boolean) メソッド

SQL 文内の ordinal で定義されたホスト変数に boolean 値を設定します。

構文

```
public void set (
    int ordinal,
    boolean value
) throws ULjException
```

パラメータ

ordinal SQL 文でのホスト変数の順序を表す 1 から始まる整数。
value 設定する値。

1.15.14.16 set(int, byte[]) メソッド

SQL 文内の ordinal で定義されたホスト変数に byte 配列値を設定します。

構文

```
public void set (
    int ordinal,
    byte[] value
) throws ULjException
```

パラメータ

ordinal SQL 文でのホスト変数の順序を表す 1 から始まる整数。
value 設定する値。

1.15.14.17 set(int, double) メソッド

SQL 文内の ordinal で定義されたホスト変数に double 値を設定します。

構文

```
public void set (
    int ordinal,
    double value
) throws ULjException
```

パラメータ

ordinal SQL 文でのホスト変数の順序を表す 1 から始まる整数。
value 設定する値。

1.15.14.18 set(int, float) メソッド

SQL 文内の ordinal で定義されたホスト変数に float 値を設定します。

構文

```
public void set (
```

```
    int ordinal,
    float value
) throws ULjException
```

パラメータ

ordinal SQL 文でのホスト変数の順序を表す 1 から始まる整数。
value 設定する値。

1.15.14.19 set(int, int) メソッド

SQL 文内の ordinal で定義されたホスト変数に int 値を設定します。

構文

```
public void set (
    int ordinal,
    int value
) throws ULjException
```

パラメータ

ordinal SQL 文でのホスト変数の順序を表す 1 から始まる整数。
value 設定する値。

1.15.14.20 set(int, long) メソッド

SQL 文内の ordinal で定義されたホスト変数に long 型の整数値を設定します。

構文

```
public void set (
    int ordinal,
    long value
) throws ULjException
```

パラメータ

ordinal SQL 文でのホスト変数の順序を表す 1 から始まる整数。
value 設定する値。

1.15.15 setNull メソッド

SQL 文内の name で定義されたホスト変数に NULL 値を設定します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public void	setNull(String) [129 ページ]	SQL 文内の name で定義されたホスト変数に NULL 値を設定します。
public void	setNull(int) [130 ページ]	SQL 文のホスト変数に NULL 値を設定します。

このセクションの内容:

setNull(String) メソッド [129 ページ]
SQL 文内の name で定義されたホスト変数に NULL 値を設定します。

setNull(int) メソッド [130 ページ]
SQL 文内の ordinal で定義されたホスト変数に NULL 値を設定します。

1.15.15.1 setNull(String) メソッド

SQL 文内の name で定義されたホスト変数に NULL 値を設定します。

構文

```
public void setNull (String name) throws ULjException
```

パラメータ

name ホスト変数名を表す文字列。

1.15.15.2 setNull(int) メソッド

SQL 文内の ordinal で定義されたホスト変数に NULL 値を設定します。

構文

```
public void setNull (int ordinal) throws ULjException
```

パラメータ

ordinal SQL 文でのホスト変数の順序を表す 1 から始まる整数。

1.16 ResultSet インタフェース

テーブルをローごとにトラバースし、カラムデータにアクセスするメソッドを提供します。

構文

```
public interface ResultSet
```

メンバー

ResultSet のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public boolean	afterLast() [133 ページ]	カーソルを最後のローの後に移動します。
public boolean	beforeFirst() [134 ページ]	カーソルを最初のローの前に移動します。
public void	close() [134 ページ]	ResultSet オブジェクトを閉じて、関連付けられているメモリリソースを解放します。
public boolean	first() [134 ページ]	カーソルを最初のローに移動します。
public java.io.InputStream	getBlobInputStream [135 ページ]	ファイルベースの long binary を含む、long binary 型の InputStream オブジェクトを返します。
public boolean	getBoolean [136 ページ]	boolean 値を返します。

変更子とタイプ	メソッド	説明
public byte[]	getBytes [138 ページ]	byte 配列を返します。
public java.io.Reader	getClobReader [139 ページ]	Reader オブジェクトを返します。
public java.util.Date	getDate [141 ページ]	java.util.Date オブジェクトを返します。
public DecimalNumber	getDecimalNumber [142 ページ]	DecimalNumber オブジェクトを返します。
public double	getDouble [144 ページ]	カラム名に基づいた double 値を返します。
public float	getFloat [146 ページ]	float 値を返します。
public int	getInt [147 ページ]	整数値を返します。
public long	getLong [149 ページ]	long 型の整数値を返します。
public int	getOrdinal(String) [150 ページ]	String で表現された値の (1 から始まる) 順序を返します。
public ResultSetMetadata	getResultSetMetadata() [151 ページ]	ResultSet オブジェクトのメタデータを含む ResultSetMetadata オブジェクトを返します。
public long	getRowCount(long) [151 ページ]	テーブルのローの数を取得します。
public int	getSize [152 ページ]	結果セットカラムの実際のサイズを取得します。
public String	getString [153 ページ]	String 値を返します。
public UUIDValue	getUUIDValue [155 ページ]	UUIDValue オブジェクトを返します。
public boolean	isNull [156 ページ]	指定されたカラム名が NULL かどうかをテストします。
public boolean	last() [158 ページ]	カーソルを最後のローに移動します。
public boolean	next() [158 ページ]	ResultSet オブジェクト内の次のデータローをフェッチします。
public boolean	previous() [159 ページ]	ResultSet オブジェクト内の前のデータローをフェッチします。
public boolean	relative(int) [159 ページ]	カーソルを、現在のカーソルの位置から、offset で指定したロー数分移動します。

備考

ResultSet オブジェクトは、SQL SELECT 文により、PreparedStatement オブジェクト上で execute または executeQuery メソッドが呼び出されたときに生成されます。

次の例は、ResultSet オブジェクトでローをフェッチし、指定したカラムのデータにアクセスする方法を示しています。

```
// Define a new SQL SELECT statement.
String sql_string = "SELECT column1, column2 FROM SampleTable";
// Create a new PreparedStatement from an existing connection.
PreparedStatement ps = conn.prepareStatement(sql_string);
// Create a new ResultSet to contain the query results of the SQL statement.
ResultSet rs = ps.executeQuery();
```

```
// Check if the PreparedStatement contains a ResultSet.
if (ps.hasResultSet()) {
    // Retrieve the column1 value from the first row using getString.
    String row1_col1 = rs.getString(1);
    // Get the next row in the table.
    if (rs.next) {
         // Retrieve the value of column1 from the second row.
         String row2_col1 = rs.getString(1);
    }
}
rs.close();
ps.close();
```

このセクションの内容:

getResultSetMetadata() メソッド [151 ページ]
ResultSet オブジェクトのメタデータを含む ResultSetMetadata オブジェクトを返します。

getRowCount(long) メソッド [151 ページ]
テーブルのローの数を取得します。

getSize メソッド [152 ページ]
結果セットカラムの実際のサイズを取得します。

getString メソッド [153 ページ]
String 値を返します。

getUUIDValue メソッド [155 ページ]
UUIDValue オブジェクトを返します。

isNull メソッド [156 ページ]
指定されたカラム名が NULL かどうかをテストします。

last() メソッド [158 ページ]
カーソルを最後のローに移動します。

next() メソッド [158 ページ]
ResultSet オブジェクト内の次のデータローをフェッチします。

previous() メソッド [159 ページ]
ResultSet オブジェクト内の前のデータローをフェッチします。

relative(int) メソッド [159 ページ]
カーソルを、現在のカーソルの位置から、offset で指定したロー数分移動します。

関連情報

PreparedStatement インタフェース [106 ページ]
execute() メソッド [109 ページ]
executeQuery() メソッド [110 ページ]
Connection インタフェース [31 ページ]

1.16.1 afterLast() メソッド

カーソルを最後のローの後に移動します。

構文

```
public boolean afterLast () throws ULjException
```

戻り値

成功した場合は true、失敗した場合は false。

1.16.2 beforeFirst() メソッド

カーソルを最初のローの前に移動します。

構文

```
public boolean beforeFirst () throws ULjException
```

戻り値

成功した場合は true、失敗した場合は false。

1.16.3 close() メソッド

ResultSet オブジェクトを閉じて、関連付けられているメモリリソースを解放します。

構文

```
public void close () throws ULjException
```

備考

以降で、閉じた ResultSet オブジェクトからローをフェッチしようとするとエラーが発生します。

1.16.4 first() メソッド

カーソルを最初のローに移動します。

構文

```
public boolean first () throws ULjException
```

戻り値

成功した場合は true、失敗した場合は false。

1.16.5 getBlobInputStream メソッド

ファイルベースの long binary を含む、long binary 型の InputStream オブジェクトを返します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public java.io.InputStream	getBlobInputStream(String) [135 ページ]	ファイルベースの long binary を含む、long binary 型の InputStream オブジェクトを返します。
public java.io.InputStream	getBlobInputStream(int) [136 ページ]	ファイルベースの long binary を含む、long binary 型の InputStream オブジェクトを返します。

このセクションの内容:

1.16.5.1 getBlobInputStream(String) メソッド

ファイルベースの long binary を含む、long binary 型の InputStream オブジェクトを返します。

構文

```
public java.io.InputStream getBlobInputStream (String name) throws ULjException
```

パラメータ

name テーブルのカラム名を表す文字列。

戻り値

指定された値の InputStream オブジェクトの表現。

1.16.5.2 getBlobInputStream(int) メソッド

ファイルベースの long binary を含む、long binary 型の InputStream オブジェクトを返します。

構文

```
public java.io.InputStream getBlobInputStream (int ordinal) throws ULjException
```

パラメータ

ordinal SQL 文でのカラムの順序を表す 1 から始まる整数。

戻り値

指定された値の InputStream オブジェクトの表現。

1.16.6 getBoolean メソッド

boolean 値を返します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public boolean	getBoolean(String) [137 ページ]	boolean 値を返します。
public boolean	getBoolean(int) [137 ページ]	boolean 値を返します。

このセクションの内容:

getBoolean(int) メソッド [137 ページ]
boolean 値を返します。

1.16.6.1 getBoolean(String) メソッド

boolean 値を返します。

構文

```
public boolean getBoolean (String name) throws ULjException
```

パラメータ

name ResultSet オブジェクトでテーブルのカラム名を表す文字列。

戻り値

指定された値の boolean 表現。

1.16.6.2 getBoolean(int) メソッド

boolean 値を返します。

構文

```
public boolean getBoolean (int ordinal) throws ULjException
```

パラメータ

ordinal SQL 文でのカラムの順序を表す 1 から始まる整数。

戻り値

指定された値の boolean 表現。

1.16.7 getBytes メソッド

byte 配列を返します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public byte[]	getBytes(String) [138 ページ]	byte 配列を返します。
public byte[]	getBytes(int) [139 ページ]	byte 配列を返します。

このセクションの内容:

1.16.7.1 getBytes(String) メソッド

byte 配列を返します。

構文

```
public byte[] getBytes (String name) throws ULjException
```

パラメータ

name テーブルのカラム名を表す文字列。

戻り値

指定された値の byte 配列表現。

1.16.7.2 getBytes(int) メソッド

byte 配列を返します。

構文

```
public byte[] getBytes (int ordinal) throws ULjException
```

パラメータ

ordinal SQL 文でのカラムの順序を表す 1 から始まる整数。

戻り値

指定された値の byte 配列表現。

1.16.8 getClobReader メソッド

Reader オブジェクトを返します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public java.io.Reader	getClobReader(String) [140 ページ]	Reader オブジェクトを返します。
public java.io.Reader	getClobReader(int) [140 ページ]	Reader オブジェクトを返します。

このセクションの内容:

getClobReader(String) メソッド [140 ページ]
Reader オブジェクトを返します。

getClobReader(int) メソッド [140 ページ]
Reader オブジェクトを返します。

1.16.8.1 getClobReader(String) メソッド

Reader オブジェクトを返します。

構文

```
public java.io.Reader getClobReader (String name) throws ULjException
```

パラメータ

name テーブルのカラム名を表す文字列。

戻り値

指定された値の Reader オブジェクト表現。

1.16.8.2 getClobReader(int) メソッド

Reader オブジェクトを返します。

構文

```
public java.io.Reader getClobReader (int ordinal) throws ULjException
```

パラメータ

ordinal SQL 文でのカラムの順序を表す 1 から始まる整数。

戻り値

指定された値の Reader オブジェクト表現。

1.16.9 getDate メソッド

java.util.Date オブジェクトを返します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public java.util.Date	getDate(String) [141 ページ]	java.util.Date オブジェクトを返します。
public java.util.Date	getDate(int) [142 ページ]	java.util.Date オブジェクトを返します。

このセクションの内容:

1.16.9.1 getDate(String) メソッド

java.util.Date オブジェクトを返します。

構文

```
public java.util.Date getDate (String name) throws ULjException
```

パラメータ

name テーブルのカラム名を表す文字列。

戻り値

指定された値の java.util.date オブジェクト表現。

1.16.9.2 getDate(int) メソッド

java.util.Date オブジェクトを返します。

構文

```
public java.util.Date getDate (int ordinal) throws ULjException
```

パラメータ

ordinal SQL 文でのカラムの順序を表す 1 から始まる整数。

戻り値

指定された値の java.util.Date オブジェクト表現。

1.16.10 getDecimalNumber メソッド

DecimalNumber オブジェクトを返します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public DecimalNumber	getDecimalNumber(String) [143 ページ]	DecimalNumber オブジェクトを返します。
public DecimalNumber	getDecimalNumber(int) [143 ページ]	DecimalNumber オブジェクトを返します。

このセクションの内容:

getDecimalNumber(int) メソッド [143 ページ]
DecimalNumber オブジェクトを返します。

1.16.10.1 getDecimalNumber(String) メソッド

DecimalNumber オブジェクトを返します。

構文

```
public DecimalNumber getDecimalNumber (String name) throws ULjException
```

パラメータ

name テーブルのカラム名を表す文字列。

戻り値

指定された値の DecimalNumber オブジェクト表現。

関連情報

DecimalNumber インタフェース [69 ページ]

1.16.10.2 getDecimalNumber(int) メソッド

DecimalNumber オブジェクトを返します。

構文

```
public DecimalNumber getDecimalNumber (int ordinal) throws ULjException
```

パラメータ

ordinal SQL 文でのカラムの順序を表す 1 から始まる整数。

戻り値

指定された値の DecimalNumber オブジェクト表現。

関連情報

DecimalNumber インタフェース [69 ページ]

1.16.11 getDouble メソッド

カラム名に基づいた double 値を返します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public double	getDouble(String) [145 ページ]	カラム名に基づいた double 値を返します。
public double	getDouble(int) [145 ページ]	カラム番号に基づいた double 値を返します。

このセクションの内容:

getDouble(String) メソッド [145 ページ]
カラム名に基づいた double 値を返します。

getDouble(int) メソッド [145 ページ]
カラム番号に基づいた double 値を返します。

1.16.11.1 getDouble(String) メソッド

カラム名に基づいた double 値を返します。

構文

```
public double getDouble (String name) throws ULjException
```

パラメータ

name テーブルのカラム名を表す文字列。

戻り値

指定された値の double 表現。

1.16.11.2 getDouble(int) メソッド

カラム番号に基づいた double 値を返します。

構文

```
public double getDouble (int ordinal) throws ULjException
```

パラメータ

ordinal SQL 文でのカラムの順序を表す 1 から始まる整数。

戻り値

指定された値の double 表現。

1.16.12 getFloat メソッド

float 値を返します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public float	getFloat(String) [146 ページ]	float 値を返します。
public float	getFloat(int) [147 ページ]	float 値を返します。

このセクションの内容:

getFloat(String) メソッド [146 ページ]
float 値を返します。

getFloat(int) メソッド [147 ページ]
float 値を返します。

1.16.12.1 getFloat(String) メソッド

float 値を返します。

構文

```
public float getFloat (String name) throws ULjException
```

パラメータ

name テーブルのカラム名を表す文字列。

戻り値

指定された値の float 表現。

1.16.12.2 getFloat(int) メソッド

float 値を返します。

構文

```
public float getFloat (int ordinal) throws ULjException
```

パラメータ

ordinal SQL 文でのカラムの順序を表す 1 から始まる整数。

戻り値

指定された値の float 表現。

1.16.13 getInt メソッド

整数値を返します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public int	getInt(String) [148 ページ]	整数値を返します。
public int	getInt(int) [148 ページ]	整数値を返します。

このセクションの内容:

getInt(String) メソッド [148 ページ]
整数値を返します。

getInt(int) メソッド [148 ページ]
整数値を返します。

1.16.13.1 getInt(String) メソッド

整数値を返します。

構文

```
public int getInt (String name) throws ULjException
```

パラメータ

name テーブルのカラム名を表す文字列。

戻り値

指定された値の整数表現。

1.16.13.2 getInt(int) メソッド

整数値を返します。

構文

```
public int getInt (int ordinal) throws ULjException
```

パラメータ

ordinal SQL 文でのカラムの順序を表す 1 から始まる整数。

戻り値

指定された値の整数表現。

1.16.14 getLong メソッド

long 型の整数値を返します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public long	getLong(String) [149 ページ]	long 型の整数値を返します。
public long	getLong(int) [150 ページ]	long 型の整数値を返します。

このセクションの内容:

getLong(String) メソッド [149 ページ]
long 型の整数値を返します。

getLong(int) メソッド [150 ページ]
long 型の整数値を返します。

1.16.14.1 getLong(String) メソッド

long 型の整数値を返します。

構文

```
public long getLong (String name) throws ULjException
```

パラメータ

name テーブルのカラム名を表す文字列。

戻り値

指定された値の long 型の整数表現。

1.16.14.2 getLong(int) メソッド

long 型の整数値を返します。

構文

```
public long getLong (int ordinal) throws ULjException
```

パラメータ

ordinal SQL 文でのカラムの順序を表す 1 から始まる整数。

戻り値

指定された値の long 型の整数表現。

1.16.15 getOrdinal(String) メソッド

String で表現された値の (1 から始まる) 順序を返します。

構文

```
public int getOrdinal (String name) throws ULjException
```

パラメータ

name テーブルのカラム名を表す文字列。

戻り値

順序の値。

1.16.16 getResultSetMetadata() メソッド

ResultSet オブジェクトのメタデータを含む ResultSetMetadata オブジェクトを返します。

構文

```
public ResultSetMetadata getResultSetMetadata () throws ULjException
```

戻り値

ResultSetMetadata オブジェクト。

1.16.17 getRowCount(long) メソッド

テーブルのローの数を取得します。

構文

```
public long getRowCount (long threshold) throws ULjException
```

パラメータ

threshold カウントするローの数の制限。0 はエラーがないことを示します。

戻り値

テーブル内のローの数。

備考

このメソッドは、"SELECT COUNT(*) FROM table" を実行するのと同じです。

1.16.18 getSize メソッド

結果セットカラムの実際のサイズを取得します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public int	getSize(String) [152 ページ]	結果セットカラムの実際のサイズを取得します。
public int	getSize(int) [153 ページ]	結果セットカラムの実際のサイズを取得します。

このセクションの内容:

getSize(String) メソッド [152 ページ]
結果セットカラムの実際のサイズを取得します。

getSize(int) メソッド [153 ページ]
結果セットカラムの実際のサイズを取得します。

1.16.18.1 getSize(String) メソッド

結果セットカラムの実際のサイズを取得します。

構文

```
public int getSize (String name) throws ULjException
```

パラメータ

name テーブルのカラム名を表す文字列。

戻り値

結果セットカラムの実際のサイズ。

1.16.18.2 getSize(int) メソッド

結果セットカラムの実際のサイズを取得します。

構文

```
public int getSize (int ordinal) throws ULjException
```

パラメータ

ordinal SQL 文でのカラムの順序を表す 1 から始まる整数。

戻り値

結果セットカラムの実際のサイズ。

1.16.19 getString メソッド

String 値を返します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public String	getString(String) [154 ページ]	String 値を返します。
public String	getString(int) [154 ページ]	String 値を返します。

このセクションの内容:

getString(String) メソッド [154 ページ]
String 値を返します。

getString(int) メソッド [154 ページ]
String 値を返します。

1.16.19.1 getString(String) メソッド

String 値を返します。

構文

```
public String getString (String name) throws ULjException
```

パラメータ

name テーブルのカラム名を表す文字列。

戻り値

指定された値の String 表現。

1.16.19.2 getString(int) メソッド

String 値を返します。

構文

```
public String getString (int ordinal) throws ULjException
```

パラメータ

ordinal SQL 文でのカラムの順序を表す 1 から始まる整数。

戻り値

指定された値の String 表現。

1.16.20 getUUIDValue メソッド

UUIDValue オブジェクトを返します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public UUIDValue	getUUIDValue(String) [155 ページ]	UUIDValue オブジェクトを返します。
public UUIDValue	getUUIDValue(int) [156 ページ]	UUIDValue オブジェクトを返します。

このセクションの内容:

getUUIDValue(String) メソッド [155 ページ]
UUIDValue オブジェクトを返します。

getUUIDValue(int) メソッド [156 ページ]
UUIDValue オブジェクトを返します。

1.16.20.1 getUUIDValue(String) メソッド

UUIDValue オブジェクトを返します。

構文

```
public UUIDValue getUUIDValue (String name) throws ULjException
```

パラメータ

name テーブルのカラム名を表す文字列。

戻り値

指定された値の UUIDValue オブジェクト表現。

1.16.20.2 getUUIDValue(int) メソッド

UUIDValue オブジェクトを返します。

構文

```
public UUIDValue getUUIDValue (int ordinal) throws ULjException
```

パラメータ

ordinal SQL 文でのカラムの順序を表す 1 から始まる整数。

戻り値

指定された値の UUIDValue オブジェクト表現。

関連情報

UUIDValue インタフェース [254 ページ]

1.16.21 isNull メソッド

指定されたカラム名が NULL かどうかをテストします。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public boolean	isNull(String) [157 ページ]	指定されたカラム名が NULL かどうかをテストします。
public boolean	isNull(int) [157 ページ]	指定されたカラムの値が NULL かどうかをテストします。

このセクションの内容:

指定されたカラム名が NULL かどうかをテストします。

isNull(int) メソッド [157 ページ]
指定されたカラムの値が NULL かどうかをテストします。

1.16.21.1 isNull(String) メソッド

指定されたカラム名が NULL かどうかをテストします。

構文

```
public boolean isNull (String name) throws ULjException
```

パラメータ

name テーブルのカラム名を表す文字列。

戻り値

値が NULL の場合は true、それ以外の場合は false。

1.16.21.2 isNull(int) メソッド

指定されたカラムの値が NULL かどうかをテストします。

構文

```
public boolean isNull (int ordinal) throws ULjException
```

パラメータ

ordinal SQL 文でのカラムの順序を表す 1 から始まる整数。

戻り値

値が NULL の場合は true、それ以外の場合は false。

1.16.22 last() メソッド

カーソルを最後のローに移動します。

構文

```
public boolean last () throws ULjException
```

戻り値

成功した場合は true、失敗した場合は false。

1.16.23 next() メソッド

ResultSet オブジェクト内の次のデータローをフェッチします。

構文

```
public boolean next () throws ULjException
```

戻り値

次のローが正常にフェッチされた場合は true、それ以外の場合は false。

関連情報

ResultSetMetadata インタフェース [160 ページ]

1.16.24 previous() メソッド

ResultSet オブジェクト内の前のデータローをフェッチします。

構文

```
public boolean previous () throws ULjException
```

戻り値

前のローが正常にフェッチされた場合は true、それ以外の場合は false。

1.16.25 relative(int) メソッド

カーソルを、現在のカーソルの位置から、offset で指定したロー数分移動します。

構文

```
public boolean relative (int offset) throws ULjException
```

パラメータ

offset 移動するローの数。

戻り値

成功した場合は true、失敗した場合は false。

1.17 ResultSetMetadata インタフェース

ResultSet オブジェクトに関連付けられ、カラム情報を提供するメソッドが含まれます。

構文

```
public interface ResultSetMetadata
```

メンバー

ResultSetMetadata のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public String	getAliasName(int) [162 ページ]	カラムのエイリアス名を返します。
public int	getColumnCount() [162 ページ]	ResultSet オブジェクト内のカラムの合計数を返します。
public String	getCorrelationName(int) [163 ページ]	カラムの相関名を返します。
public String	getDomainName(int) [163 ページ]	ドメインの名前を返します。
public int	getDomainPrecision(int) [164 ページ]	ドメイン値の精度を返します。
public int	getDomainScale(int) [164 ページ]	ドメイン値の位取りを返します。
public int	getDomainSize(int) [165 ページ]	ドメイン値のサイズを返します。
public short	getDomainType(int) [165 ページ]	ドメインの型を返します。
public String	getQualifiedName(int) [166 ページ]	カラムの修飾名を返します。
public String	getTableColumnName(int) [166 ページ]	テーブルまたは派生テーブルのカラム名を返します。
public String	getTableName(int) [167 ページ]	カラムのテーブル名を返します。
public String	getWrittenName(int) [168 ページ]	カラムの書き込み名を返します。

備考

このインタフェースは、ResultSet.getResultSetMetadata メソッドによって取得されます。

ResultSet オブジェクトの選択リストのカラムが簡略名または複合名 (table-name.column-name、correlation-name.column-name) の場合、名前に関する次の情報が存在すると、その情報が抽出されます。

- エイリアス名
- 相関名

- 名前の修飾バージョン
- テーブル名
- 書き込まれる名前

ResultSet オブジェクトの選択リストのカラムごとに、そのカラムのドメインに関する次の情報を取得できます。

- カラムの型: Domain インタフェースの整数
- ドメインの名前
- ドメインのサイズ (VARCHAR ドメインと BINARY ドメインの場合)
- 位取りと精度 (NUMERIC ドメインの場合)

このセクションの内容:

関連情報

Domain インタフェース [74 ページ]
ResultSet インタフェース [130 ページ]

getResultSetMetadata() メソッド [151 ページ]

1.17.1 getAliasName(int) メソッド

カラムのエイリアス名を返します。

構文

```
public String getAliasName (int column_no) throws ULjException
```

パラメータ

column_no 選択リストの (1 から始まる) カラム番号。

戻り値

カラムにエイリアス名がない場合は NULL、そうでない場合はカラムのエイリアス名を返します。

備考

エイリアス名 ([AS] 名) は、カラムを参照するために指定できます。

1.17.2 getColumnCount() メソッド

ResultSet オブジェクト内のカラムの合計数を返します。

構文

```
public int getColumnCount () throws ULjException
```

戻り値

カラムの数。

1.17.3 getCorrelationName(int) メソッド

カラムの相関名を返します。

構文

```
public String getCorrelationName (int column_no) throws ULjException
```

パラメータ

column_no 選択リストの (1 から始まる) カラム番号。

戻り値

カラムに相関名がない場合は NULL、そうでない場合はカラムの相関名を返します。

備考

FROM 句で指定した相関名 ([AS] correlation-name) により、派生テーブルなどのテーブル表現を指定します。

1.17.4 getDomainName(int) メソッド

ドメインの名前を返します。

構文

```
public String getDomainName (int column_no) throws ULjException
```

パラメータ

column_no 選択リストの (1 から始まる) カラム番号。

戻り値

ドメイン名。

1.17.5 getDomainPrecision(int) メソッド

ドメイン値の精度を返します。

構文

```
public int getDomainPrecision (int column_no) throws ULjException
```

パラメータ

column_no 選択リストの (1 から始まる) カラム番号。

戻り値

精度。

1.17.6 getDomainScale(int) メソッド

ドメイン値の位取りを返します。

構文

```
public int getDomainScale (int column_no) throws ULjException
```

パラメータ

column_no 選択リストの (1 から始まる) カラム番号。

戻り値

位取り。

1.17.7 getDomainSize(int) メソッド

ドメイン値のサイズを返します。

構文

```
public int getDomainSize (int column_no) throws ULjException
```

パラメータ

column_no 選択リストの (1 から始まる) カラム番号。

戻り値

サイズ。

1.17.8 getDomainType(int) メソッド

ドメインの型を返します。

構文

```
public short getDomainType (int column_no) throws ULjException
```

パラメータ

column_no 選択リストの (1 から始まる) カラム番号。

戻り値

整数で表したドメインの型。

1.17.9 getQualifiedName(int) メソッド

カラムの修飾名を返します。

構文

```
public String getQualifiedName (int column_no) throws ULjException
```

パラメータ

column_no 選択リストの (1 から始まる) カラム番号。

戻り値

カラムに修飾名がない場合は NULL、そうでない場合はカラムの修飾名を返します。

備考

ResultSet カラムがテーブルのカラムを参照している場合、相関名 (相関名が指定されていない場合はテーブル名) の後にテーブルのカラム名が続く複合名が返されます。

ResultSet カラムがテーブルのカラムを参照していない場合、エイリアスが指定されていれば、エイリアス名が返されます。

1.17.10 getTableColumnName(int) メソッド

テーブルまたは派生テーブルのカラム名を返します。

構文

```
public String getTableColumnName (int column_no) throws ULjException
```

パラメータ

column_no 選択リストの (1 から始まる) カラム番号。

戻り値

カラムにテーブル名がない場合は NULL、そうでない場合はカラムのテーブル名を返します。

1.17.11 getTableName(int) メソッド

カラムのテーブル名を返します。

構文

```
public String getTableName (int column_no) throws ULjException
```

パラメータ

column_no 選択リストの (1 から始まる) カラム番号。

戻り値

カラムにテーブル名がない場合は NULL、そうでない場合はカラムのテーブル名を返します。

備考

テーブルは、ResultSet カラムが参照するテーブル名 (相関名の可能性あり) です。

1.17.12 getWrittenName(int) メソッド

カラムの書き込み名を返します。

構文

```
public String getWrittenName (int column_no) throws ULjException
```

パラメータ

column_no 選択リストの (1 から始まる) カラム番号。

戻り値

カラムに書き込み名がない場合は NULL、そうでない場合はカラムの書き込み名を返します。

備考

書き込み名は、選択リストに指示カラムとして指定された単純名または複合名です。

1.18 SQLInfo インタフェース

実行された SQL 文についての情報を示します。

構文

```
public interface SQLInfo
```

メンバー

SQLInfo のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public String	getMessage() [169 ページ]	SQL コードに関連付けられたメッセージを返します。
public String	getParameter(short) [169 ページ]	指定されたパラメータを返します。
public short	getParameterCount() [170 ページ]	メッセージのパラメータの数を返します。
public int	getSQLCode() [170 ページ]	実行された SQL 文のコード (SQLCODE) を取得します。
public int	getSQLCount() [171 ページ]	実行された SQL 文の結果によって異なる値を返します。

このセクションの内容:

1.18.1 getMessage() メソッド

SQL コードに関連付けられたメッセージを返します。

構文

```
public String getMessage ()
```

1.18.2 getParameter(short) メソッド

指定されたパラメータを返します。

構文

```
public String getParameter (short param_no)
```

パラメータ

param_no 1 から始まるパラメータ番号。

戻り値

パラメータ。

1.18.3 getParameterCount() メソッド

メッセージのパラメータの数を返します。

構文

```
public short getParameterCount ()
```

戻り値

パラメータ数。

1.18.4 getSQLCode() メソッド

実行された SQL 文のコード (SQLCODE) を取得します。

構文

```
public int getSQLCode ()
```

戻り値

SQLCODE 値。

1.18.5 getSQLCount() メソッド

実行された SQL 文の結果によって異なる値を返します。

構文

```
public int getSQLCount ()
```

戻り値

INSERT、UPDATE、または DELETE 文の実行後に、文の影響を受けたローの数。SQLE_SYNTAX_ERROR が発生した場合の戻り値は、エラーに該当する動的 SQL 文のオフセットです。

1.19 StreamHTTPParms インタフェース

HTTP を使用して Mobile Link サーバと通信する方法を定義する HTTP ストリームパラメータを表します。

構文

```
public interface StreamHTTPParms extends StreamTCPIPParms
```

メンバー

StreamHTTPParms のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public String	getE2eePublicKey() [174 ページ]	エンドツーエンドのパブリックキーを含むファイルの名前を返します。
public String	getExtraParameters() [174 ページ]	予備の Mobile Link クライアントネットワークプロトコルオプションを取得します。
public String	getHost() [175 ページ]	Mobile Link サーバのホスト名を返します。
public int	getOutputBufferSize() [175 ページ]	データが Mobile Link サーバに送信される前に格納される出力バッファのサイズをバイト単位で返します。

変更子とタイプ	メソッド	説明
public int	getPort() [176 ページ]	Mobile Link サーバへの接続に使用されているポート番号を返します。
public String	getURLSuffix() [176 ページ]	Mobile Link サーバの URL サフィックスを返します。
public boolean	isRestartable() [177 ページ]	再起動可能な HTTP が使用されているかどうかを判断します。
public void	setE2eePublicKey(String) [177 ページ]	エンドツーエンドのパブリックキーを含むファイルの名前を指定します。
public void	setExtraParameters(String) [178 ページ]	予備の Mobile Link クライアントネットワークプロトコルオプションを設定します。
public void	setHost(String) [179 ページ]	Mobile Link サーバのホスト名を設定します。
public void	setOutputBufferSize(int) [179 ページ]	データが Mobile Link サーバに送信される前に格納される出力バッファのサイズをバイト単位で設定します。
public void	setPort(int) [180 ページ]	Mobile Link サーバへの接続に使用するポート番号を設定します。
public void	setRestartable(boolean) [181 ページ]	再起動可能な HTTP を有効または無効にします。
public void	setURLSuffix(String) [181 ページ]	Mobile Link サーバに接続するための URL サフィックスを指定します。
public void	setZlibCompression(boolean) [182 ページ]	ZLIB 圧縮を有効または無効にします。
public void	setZlibDownloadWindowSize(int) [183 ページ]	ZLIB 圧縮のダウンロードウィンドウサイズを設定します。
public void	setZlibUploadWindowSize(int) [183 ページ]	ZLIB 圧縮のアップロードウィンドウサイズを設定します。
public boolean	zlibCompressionEnabled() [184 ページ]	ZLIB 圧縮が有効かどうかを判断します。

備考

次の例では、ホスト名 "MyMLHost" にある Mobile Link サーバと通信するようにストリームパラメータを設定しています。サーバは次のパラメータで起動されました。"-x http(port=1234)":

```
SyncParms syncParms = myConnection.createSyncParms(
   SyncParms.HTTP_STREAM,
   "MyUniqueMLUserID",
   "MyMLScriptVersion"
);
StreamHTTPParms httpParms = syncParms.getStreamParms();
httpParms.setHost("MyMLHost");
httpParms.setPort(1234);
```

このインタフェースを実装するインスタンスは、SyncParms.getStreamParms メソッドで返されます。

このセクションの内容:

関連情報

SyncParms クラス [198 ページ]
getStreamParms() メソッド [207 ページ]

1.19.1 getE2eePublicKey() メソッド

エンドツーエンドのパブリックキーを含むファイルの名前を返します。

構文

```
public String getE2eePublicKey ()
```

戻り値

エンドツーエンドのパブリックキーを含むファイルの名前。

関連情報

setE2eePublicKey(String) メソッド [177 ページ]

1.19.2 getExtraParameters() メソッド

予備の Mobile Link クライアントネットワークプロトコルオプションを取得します。

構文

```
public String getExtraParameters ()
```

戻り値

設定された予備のプロトコルオプション。

1.19.3 getHost() メソッド

Mobile Link サーバのホスト名を返します。

構文

```
public String getHost ()
```

戻り値

ホスト名。

関連情報

setHost(String) メソッド [179 ページ]
getPort() メソッド [176 ページ]
setPort(int) メソッド [180 ページ]

1.19.4 getOutputBufferSize() メソッド

データが Mobile Link サーバに送信される前に格納される出力バッファのサイズをバイト単位で返します。

構文

```
public int getOutputBufferSize ()
```

戻り値

バッファサイズの整数値。

備考

この値を大きくすると、サイズの大きいアップロードの送信に必要なネットワークフラッシュの回数が減る可能性がありますが、メモリの使用量が増えます。HTTP では、フラッシュごとに大容量 (約 250 バイト) の HTTP ヘッダが送信されるので、フラッシュの回数が減ると帯域幅の使用量が削減されます。

関連情報

setOutputBufferSize(int) メソッド [179 ページ]

1.19.5 getPort() メソッド

Mobile Link サーバへの接続に使用されているポート番号を返します。

構文

```
public int getPort ()
```

戻り値

Mobile Link サーバのポート番号。

関連情報

setPort(int) メソッド [180 ページ]

1.19.6 getURLSuffix() メソッド

Mobile Link サーバの URL サフィックスを返します。

構文

```
public String getURLSuffix ()
```

戻り値

URL サフィックスを含む文字列。

関連情報

setURLSuffix(String) メソッド [181 ページ]

1.19.7 isRestartable() メソッド

再起動可能な HTTP が使用されているかどうかを判断します。

構文

```
public boolean isRestartable ()
```

戻り値

再起動可能な HTTP が有効である場合は true、それ以外は false。

関連情報

setRestartable(boolean) メソッド [181 ページ]

1.19.8 setE2eePublicKey(String) メソッド

エンドツーエンドのパブリックキーを含むファイルの名前を指定します。

構文

```
public void setE2eePublicKey (String public_key)
```

パラメータ

public_key 暗号化に使用する RSA パブリックキーファイルの名前。この名前は DER でコード化される必要があります。

備考

デフォルトでは、この値は NULL で、エンドツーエンドの暗号化は使用されないことを示します。

このメソッドは e2ee_public_key プロトコルオプションに対応します。

パブリックキーは、SD カードまたはデバイスのオブジェクトストアに格納できます。

SD カードを使用する場合、public_key パラメータは次の形式にしてください。

file://path

path はカード上のこのファイルへの絶対パスです。たとえば、`file:///SDCard/ulj/public_key.der` は有効な public_key パラメータです。

オブジェクトストアを使用する場合は、Ultra Light Java Edition Database Transfer ユーティリティを使用して Mobile Link サーバからファイルをダウンロードします。キーには、`der` ファイル拡張子が必要です。

関連情報

getE2eePublicKey() メソッド [174 ページ]

1.19.9 setExtraParameters(String) メソッド

予備の Mobile Link クライアントネットワークプロトコルオプションを設定します。

構文

```
public void setExtraParameters (String parms)
```

パラメータ

parms セミコロンで区切ったプロトコルオプションのリスト。

備考

これらのオプションは、このクラスのメソッドの結果である設定から構築されるリストに付加されます。

このメソッドによって設定されるオプションにより、他のメソッドによって設定された同じオプションは上書きされます。たとえば、予備のパラメータに "host=abc" が含まれている場合、setHost("xyz") メソッドが呼び出されると、ホストオプションは "abc" になります。

1.19.10 setHost(String) メソッド

Mobile Link サーバのホスト名を設定します。

構文

```
public void setHost (String v)
```

パラメータ

v ホスト名。

備考

デフォルトは NULL で、これは localhost を示します。

関連情報

getHost() メソッド [175 ページ]
getPort() メソッド [176 ページ]
setPort(int) メソッド [180 ページ]

1.19.11 setOutputBufferSize(int) メソッド

データが Mobile Link サーバに送信される前に格納される出力バッファのサイズをバイト単位で設定します。

構文

```
public void setOutputBufferSize (int size)
```

パラメータ

size 新しいバッファのサイズ。

備考

デフォルトは 4096 です。有効な値の範囲は 512 ～ 32768 です。この値を大きくすると Java ランタイムから、Mobile Link サーバでは処理できないチャンク形式の HTTP が送信される可能性があります。

Mobile Link サーバから "未知の転送エンコードです" というエラーが出力された場合は、この値を小さくしてみてください。

関連情報

getOutputBufferSize() メソッド [175 ページ]

1.19.12 setPort(int) メソッド

Mobile Link サーバへの接続に使用するポート番号を設定します。

構文

```
public void setPort (int v)
```

パラメータ

v 1 ～ 65535 のポート番号。範囲外の値はデフォルト値に変更されます。

備考

デフォルトのポートは HTTP 同期の場合は 80、HTTPS 同期の場合は 443 です。

関連情報

getPort() メソッド [176 ページ]

1.19.13 setRestartable(boolean) メソッド

再起動可能な HTTP を有効または無効にします。

構文

```
public void setRestartable (boolean isRestartable)
```

パラメータ

isRestartable 再起動可能な HTTP を有効にする場合は、true に設定します。デフォルト値は false です。

備考

再開可能な HTTP が有効になっている場合、Ultra Light J では信頼性の低いネットワークでネットワークが頻繁に失敗しないよう、ネットワークの割り込みが許容されます。

再起動可能な HTTP を使用するには、Ultra Light J と Mobile Link サーバの両方に CR#690250 が適用されている必要があります。

関連情報

isRestartable() メソッド [177 ページ]

1.19.14 setURLSuffix(String) メソッド

Mobile Link サーバに接続するための URL サフィックスを指定します。

構文

```
public void setURLSuffix (String v)
```

パラメータ

v URL サフィックスの文字列。

備考

Ultra Light J は次のフォーマットの URL を形成します。

```
[http|https]://host-name:port-number/url-suffix
```

デフォルトでは、*url-suffix* は "Mobilink/" です。**v** を NULL に設定することによって、URL サフィックスをデフォルトに設定できます。

関連情報

getURLSuffix() メソッド [176 ページ]

1.19.15 setZlibCompression(boolean) メソッド

ZLIB 圧縮を有効または無効にします。

構文

```
public void setZlibCompression (boolean enable)
```

パラメータ

enable ZLIB 圧縮を有効にする場合は true、無効にする場合は false に設定します。

備考

デフォルトでは、ZLIB 圧縮は無効です。

このメソッドは compression=zlib プロトコルオプションに対応します。

1.19.16 setZlibDownloadWindowSize(int) メソッド

ZLIB 圧縮のダウンロードウィンドウサイズを設定します。

構文

```
public void setZlibDownloadWindowSize (int size)
```

パラメータ

size 圧縮ウィンドウサイズの指定。このパラメータは、ウィンドウサイズ (履歴バッファのサイズ) を底が 2 の対数で指定します。9 ～ 15 (両端の値を含む) の有効範囲を指定します。

備考

このメソッドは、zlib_download_window_size プロトコルオプションに対応します。

1.19.17 setZlibUploadWindowSize(int) メソッド

ZLIB 圧縮のアップロードウィンドウサイズを設定します。

構文

```
public void setZlibUploadWindowSize (int size)
```

パラメータ

size 圧縮ウィンドウサイズの指定。このパラメータは、ウィンドウサイズ (履歴バッファのサイズ) を底が 2 の対数で指定します。9 ～ 15 (両端の値を含む) の有効範囲を指定します。

備考

このメソッドは、zlib_upload_window_size プロトコルオプションに対応します。

1.19.18 zlibCompressionEnabled() メソッド

ZLIB 圧縮が有効かどうかを判断します。

構文

```
public boolean zlibCompressionEnabled ()
```

戻り値

有効である場合は true、それ以外の場合は false。

関連情報

setZlibCompression(boolean) メソッド [182 ページ]

1.20 StreamHTTPSParms インタフェース

セキュア HTTPS 接続を使用して Mobile Link サーバと通信する方法を定義する HTTPS ストリームパラメータを表します。

構文

```
public interface StreamHTTPSParms extends StreamHTTPParms
```

メンバー

StreamHTTPSParms のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public String	getCertificateCompany() [187 ページ]	セキュア接続の検証に使用する証明書の会社名を返します。
public String	getCertificateName() [188 ページ]	セキュア接続の検証に使用する証明書の通称を返します。

変更子とタイプ	メソッド	説明
public String	getCertificateUnit() [188 ページ]	セキュア接続の検証に使用する証明書に記載される部署名を返します。
public String	getTrustedCertificates() [188 ページ]	安全な同期に使用される信頼できるルート証明書リストのファイル名を返します。
public void	setCertificateCompany(String) [189 ページ]	セキュア接続の検証に使用する証明書の会社名を設定します。
public void	setCertificateName(String) [189 ページ]	セキュア接続の検証に使用する証明書の通称を設定します。
public void	setCertificateUnit(String) [190 ページ]	セキュア接続の検証に使用する証明書に記載される部署名を設定します。
public void	setTrustedCertificates(String) [190 ページ]	安全な同期に使用される信頼できるルート証明書リストのファイルを設定します。

StreamHTTPParms から継承されたメンバー

変更子とタイプ	メンバー	説明
public String	getE2eePublicKey() [174 ページ]	エンドツーエンドのパブリックキーを含むファイルの名前を返します。
public String	getExtraParameters() [174 ページ]	予備の Mobile Link クライアントネットワークプロトコルオプションを取得します。
public String	getHost() [175 ページ]	Mobile Link サーバのホスト名を返します。
public int	getOutputBufferSize() [175 ページ]	データが Mobile Link サーバに送信される前に格納される出力バッファのサイズをバイト単位で返します。
public int	getPort() [176 ページ]	Mobile Link サーバへの接続に使用されているポート番号を返します。
public String	getURLSuffix() [176 ページ]	Mobile Link サーバの URL サフィックスを返します。
public boolean	isRestartable() [177 ページ]	再起動可能な HTTP が使用されているかどうかを判断します。
public void	setE2eePublicKey(String) [177 ページ]	エンドツーエンドのパブリックキーを含むファイルの名前を指定します。
public void	setExtraParameters(String) [178 ページ]	予備の Mobile Link クライアントネットワークプロトコルオプションを設定します。
public void	setHost(String) [179 ページ]	Mobile Link サーバのホスト名を設定します。
public void	setOutputBufferSize(int) [179 ページ]	データが Mobile Link サーバに送信される前に格納される出力バッファのサイズをバイト単位で設定します。
public void	setPort(int) [180 ページ]	Mobile Link サーバへの接続に使用するポート番号を設定します。

変更子とタイプ	メンバー	説明
public void	setRestartable(boolean) [181 ページ]	再起動可能な HTTP を有効または無効にします。
public void	setURLSuffix(String) [181 ページ]	Mobile Link サーバに接続するための URL サフィックスを指定します。
public void	setZlibCompression(boolean) [182 ページ]	ZLIB 圧縮を有効または無効にします。
public void	setZlibDownloadWindowSize(int) [183 ページ]	ZLIB 圧縮のダウンロードウィンドウサイズを設定します。
public void	setZlibUploadWindowSize(int) [183 ページ]	ZLIB 圧縮のアップロードウィンドウサイズを設定します。
public boolean	zlibCompressionEnabled() [184 ページ]	ZLIB 圧縮が有効かどうかを判断します。

備考

次の例では、ホスト名 "MyMLHost" にある Mobile Link サーバと通信するようにストリームパラメータを設定しています。サーバは次のパラメータで起動されました。"-x https(port=1234;identity=RSAServer.id;identity_password=x)"

```
SyncParms syncParms = myConnection.createSyncParms(
       SyncParms.HTTPS_STREAM,
       "MyUniqueMLUserID",
       "MyMLScriptVersion"
   );
StreamHTTPSParms httpsParms =
   (StreamHTTPSParms) syncParms.getStreamParms();
httpsParms.setHost("MyMLHost");
httpsParms.setPort(1234);
```

上記の例では、RSAServer.id 内の ID が、クライアントホストまたはデバイスにインストール済みの信頼できるルート ID のチェーンに追加されていることを前提としています。

J2SE の場合、次のいずれかの方法で、必要な信用されたルート ID を配備できます。

1. 信頼できるルート証明書 ID を JRE の lib/security/cacerts キーストアにインストールします。
2. Java の keytool ユーティリティを使用して独自のキーストアを構築し、Java システムプロパティ javax.net.ssl.trustStore をその場所に設定する (javax.net.ssl.trustStorePassword メソッドを適切な値に設定する)。
3. setTrustedCertificates(String) パラメータを使用して、配備された ID ファイルを参照します。

セキュリティを強化するには、setCertificateName、setCertificateCompany、setCertificateUnit の各メソッドを使用して Mobile Link サーバの ID の検証を有効にします。

このインタフェースを実装するインスタンスは、HTTPS 同期用に SyncParms オブジェクトが作成されたときに、SyncParms.getStreamParms メソッドによって返されます。

このセクションの内容:

getCertificateCompany() メソッド [187 ページ]
セキュア接続の検証に使用する証明書の会社名を返します。

getCertificateName() メソッド [188 ページ]
セキュア接続の検証に使用する証明書の通称を返します。

getCertificateUnit() メソッド [188 ページ]
セキュア接続の検証に使用する証明書に記載される部署名を返します。

getTrustedCertificates() メソッド [188 ページ]
安全な同期に使用される信頼できるルート証明書リストのファイル名を返します。

setCertificateCompany(String) メソッド [189 ページ]
セキュア接続の検証に使用する証明書の会社名を設定します。

setCertificateName(String) メソッド [189 ページ]
セキュア接続の検証に使用する証明書の通称を設定します。

setCertificateUnit(String) メソッド [190 ページ]
セキュア接続の検証に使用する証明書に記載される部署名を設定します。

setTrustedCertificates(String) メソッド [190 ページ]
安全な同期に使用される信頼できるルート証明書リストのファイルを設定します。

関連情報

SyncParms クラス [198 ページ]
getStreamParms() メソッド [207 ページ]
setCertificateCompany(String) メソッド [189 ページ]
setCertificateName(String) メソッド [189 ページ]
setCertificateUnit(String) メソッド [190 ページ]
setTrustedCertificates(String) メソッド [190 ページ]

1.20.1 getCertificateCompany() メソッド

セキュア接続の検証に使用する証明書の会社名を返します。

構文

```
public String getCertificateCompany ()
```

戻り値

証明書の会社名。

1.20.2 getCertificateName() メソッド

セキュア接続の検証に使用する証明書の通称を返します。

構文

```
public String getCertificateName ()
```

戻り値

証明書の名前。

1.20.3 getCertificateUnit() メソッド

セキュア接続の検証に使用する証明書に記載される部署名を返します。

構文

```
public String getCertificateUnit ()
```

戻り値

組織単位名。

1.20.4 getTrustedCertificates() メソッド

安全な同期に使用される信頼できるルート証明書リストのファイル名を返します。

構文

```
public String getTrustedCertificates ()
```

戻り値

信用されたルート証明書ファイルのファイル名。

関連情報

setTrustedCertificates(String) メソッド [190 ページ]

1.20.5 setCertificateCompany(String) メソッド

セキュア接続の検証に使用する証明書の会社名を設定します。

構文

```
public void setCertificateCompany (String val)
```

パラメータ

val 会社名。

備考

デフォルトは NULL で、この場合、証明書で会社名は検証されません。

1.20.6 setCertificateName(String) メソッド

セキュア接続の検証に使用する証明書の通称を設定します。

構文

```
public void setCertificateName (String val)
```

パラメータ

val 証明書の通称。

備考

デフォルトは NULL で、この場合、証明書で通称は検証されません。

1.20.7 setCertificateUnit(String) メソッド

セキュア接続の検証に使用する証明書に記載される部署名を設定します。

構文

```
public void setCertificateUnit (String val)
```

パラメータ

val 会社の組織単位名。

備考

デフォルトは NULL で、この場合、証明書で組織単位名は検証されません。

1.20.8 setTrustedCertificates(String) メソッド

安全な同期に使用される信頼できるルート証明書リストのファイルを設定します。

構文

```
public void setTrustedCertificates (String filename) throws ULjException
```

パラメータ

filename 信用されたルート証明書のファイル名。

備考

このメソッドは、プラットフォーム上の Java Runtime Environment で許可される任意の X.509 形式をサポートします。BKS KeyStore が使用されます。

証明書は、次の優先度の規則に従って使用されます。

1. このメソッドが呼び出された場合、指定したファイルの証明書が使用されます。
2. このメソッドが呼び出されず、ulinit または ulload ユーティリティによって証明書がデータベースが設定されている場合、それらの証明書が使用されます。
3. このメソッド、ulinit または ulload ユーティリティのいずれによっても証明書が指定されない場合、証明書はオペレーティングシステムの信頼できる証明書ストアから読み込まれます。この証明書ストアは、Web サーバを安全に管理するために HTTPS を介して接続するときに、Web ブラウザで使用されます。

関連情報

getTrustedCertificates() メソッド [188 ページ]

1.21 ストリーム TCPIPParms インタフェース

TCP/IP を使用して Mobile Link サーバと通信する方法を定義する TCP/IP ストリームパラメータを表します。

構文

```
public interface StreamTCPIPParms
```

メンバー

StreamHTTPSParms のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public String	getE2eePublicKey() [174 ページ]	エンドツーエンドのパブリックキーを含むファイルの名前を返します。
public String	getHost() [175 ページ]	Mobile Link サーバのホスト名を返します。
public String	getPort() [176 ページ]	Mobile Link サーバへの接続に使用されているポート番号を返します。

変更子とタイプ	メソッド	説明
public void	setE2eePublicKey(String) [177 ページ]	エンドツーエンドのパブリックキーを含むファイルの名前を指定します。
public void	setExtraParameters(String) [178 ページ]	予備の Mobile Link クライアントネットワークプロトコルオプションを設定します。
public void	setHost(String) [179 ページ]	Mobile Link サーバのホスト名を設定します。
public void	setPort(int) [180 ページ]	Mobile Link サーバへの接続に使用するポート番号を設定します。
public void	setZlibCompression(boolean) [182 ページ]	ZLIB 圧縮を有効または無効にします。
public void	setZlibDownloadWindowSize(int) [183 ページ]	ZLIB 圧縮のダウンロードウィンドウサイズを設定します。
public void	setZlibUploadWindowSize(int) [183 ページ]	ZLIB 圧縮のアップロードウィンドウサイズを設定します。
public void	zlibCompressionEnabled() method [184 ページ]	ZLIB 圧縮が有効かどうかを判断します。

備考

次の例では、ホスト名 "MyMLHost" にある Mobile Link サーバと通信するようにストリームパラメータを設定しています。サーバは次のパラメータで起動されました。"-x tcpip(port=1234)":

```
SyncParms syncParms = myConnection.createSyncParms(
    SyncParms.TCPIP_STREAM,
    "MyUniqueMLUserID",
    "MyMLScriptVersion"
);
StreamTCPIPParms sparms = syncParms.getStreamParms();
sparms.setHost("MyMLHost");
sParms.setPort(1234);
```

このインタフェースを実装するインスタンスは、SyncParms.getStreamParms メソッドで返されます。

1.22 StreamTLSParms インタフェース

セキュア TLS 接続を使用して Mobile Link サーバと通信する方法を定義する TLS ストリームパラメータを表します。

構文

```
public interface StreamTLSParms
```

メンバー

StreamHTTPSParms のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public String	getCertificateCompany() [187 ページ]	セキュア接続の検証に使用する証明書の会社名を返します。
public String	getCertificateName() [188 ページ]	セキュア接続の検証に使用する証明書の通称を返します。
public String	getCertificateUnit() [188 ページ]	セキュア接続の検証に使用する証明書に記載される部署名を返します。
public String	getTrustedCertificates() [188 ページ]	安全な同期に使用される信頼できるルート証明書リストのファイル名を返します。
public void	setCertificateCompany(String) [189 ページ]	セキュア接続の検証に使用する証明書の会社名を設定します。
public void	setCertificateName(String) [189 ページ]	セキュア接続の検証に使用する証明書の通称を設定します。
public void	setCertificateUnit(String) [190 ページ]	セキュア接続の検証に使用する証明書に記載される部署名を設定します。
public void	setTrustedCertificates(String) [190 ページ]	安全な同期に使用される信頼できるルート証明書リストのファイルを設定します。

備考

次の例では、ホスト名 "MyMLHost" にある Mobile Link サーバと通信するようにストリームパラメータを設定しています。サーバは次のパラメータで起動されました。"-x tls(port=1234;identity=RSAServer.id;identity_password=x)":

```
SyncParms syncParms = myConnection.createSyncParms(
        SyncParms.TLS_STREAM,
        "MyUniqueMLUserID",
        "MyMLScriptVersion"
    );
StreamTLSParms sParms =
    (StreamTLSParms) syncParms.getStreamParms();
sParms.setHost("MyMLHost");
sParms.setPort(1234);
```

上記の例では、RSAServer.id 内の ID が、クライアントホストまたはデバイスにインストール済みの信頼できるルート ID のチェーンに追加されていることを前提としています。

J2SE の場合、次のいずれかの方法で、必要な信用されたルート ID を配備できます。

1. 信頼できるルート証明書 ID を JRE の lib/security/cacerts キーストアにインストールします。
2. Java の keytool ユーティリティを使用して独自のキーストアを構築し、Java システムプロパティ javax.net.ssl.trustStore をその場所に設定する (javax.net.ssl.trustStorePassword メソッドを適切な値に設定する)。
3. setTrustedCertificates(String) パラメータを使用して、配備された ID ファイルを参照します。

セキュリティを強化するには、setCertificateName、setCertificateCompany、setCertificateUnit の各メソッドを使用して Mobile Link サーバの ID の検証を有効にします。

このインタフェースを実装するインスタンスは、TLS 同期用に SyncParms オブジェクトが作成されたときに、SyncParms.getStreamParms メソッドによって返されます。

1.23 SyncObserver インタフェース

同期の進行状況の情報を受け取ります。

構文

```
public interface SyncObserver
```

メンバー

SyncObserver のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public boolean	syncProgress(int, SyncResult) [195 ページ]	ユーザに進行状況を通知します。

備考

同期の進行状況レポートを受け取るには、同期を実行する新しいクラスを作成し、SyncParms.setSyncObserver メソッドを使用して実装します。

次に、SyncObserver オブジェクトの簡単な実装例を示します。

```
class MyObserver implements SyncObserver {
 public boolean syncProgress(int state, SyncResult result) {
   System.out.println(
       "sync progress state = " + state
       + " bytes sent = " + result.getSentByteCount()
       + " bytes received = " + result.getReceivedByteCount()
     );
   return false;   // Always continue synchronization.
 }
 public MyObserver() {} // The default constructor.
}
```

上記のクラスは、次のメソッドの呼び出しによって有効にできます。

```
SyncParms.setSyncObserver(new MyObserver());
```

このセクションの内容:

関連情報

SyncParms クラス [198 ページ]
setSyncObserver(SyncObserver) メソッド [221 ページ]

1.23.1 syncProgress(int, SyncResult) メソッド

ユーザに進行状況を通知します。

構文

```
public boolean syncProgress (
    int state,
    SyncResult data
)
```

パラメータ

state 同期の現在のステータスを表す SyncOberver.States 定数の 1 つ。
data 同期の最新の結果を含む SyncResult オブジェクト。

戻り値

同期をキャンセルする場合は true、続行する場合は false を返します。

備考

このメソッドは同期中に呼び出されます。

パケットとして通知されるさまざまなステータスが送受信されます。1 つのパケットで複数のテーブルがアップロードまたはダウンロードされることがあるので、任意の同期に対するこのメソッドの呼び出しでは、いくつかのステータスが省略される場合があります。

注記

SyncResult メソッドを除き、syncProgress 呼び出し中に他の Ultra Light J API メソッドを呼び出すことはできません。

関連情報

SyncObserver.States インタフェース [196 ページ]
setSyncObserver(SyncObserver) メソッド [221 ページ]
SyncResult クラス [224 ページ]

1.24 SyncObserver.States インタフェース

observer に通知できる同期ステータスを定義します。

構文

```
public interface States
```

メンバー

States のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

変数

変更子とタイプ	変数	説明
public final int	STARTING	同期を開始していることを示します。 処理はまだ行われていません。
public final int	CONNECTING	同期を開始していることを示します。 処理はまだ行われていません。
public final int	RESUMING_DOWNLOAD	部分ダウンロードを再開しようとしたときに入るオプションの状態。 成功すると、同期が UL_SYNC_STATE_RECEIVING_TABLE 状態に進みます。再開できない場合は、UL_SYNC_STATE_ERROR になります。

変更子とタイプ	変数	説明
public final int	SENDING_HEADER	同期ストリームが開かれ、ヘッダが送信されようとしています。
public final int	SENDING_CHECK_SYNC_REQUEST	前回のアップロードの状態が不明なため、ステータスをチェックする要求が送信されます。
public final int	WAITING_FOR_CHECK_SYNC_RESPONSE	サーバが同期チェック要求に応答するのを待機しています。
public final int	PROCESSING_CHECK_SYNC_RESPONSE	同期チェック要求への応答が受け取られ、処理されています。
public final int	SENDING_TABLE	新しいテーブルがアップロードされていることを示します。
public final int	SENDING_DATA	スキーマ情報またはローデータが送信されていることを示します。
public final int	FINISHING_UPLOAD	アップロードの完了処理中であることを示します。
public final int	WAITING_FOR_UPLOAD_ACK	サーバがアップロードの受信を確認するのを待機しています。
public final int	PROCESSING_UPLOAD_ACK	サーバがアップロードの受信を確認しました。
public final int	WAITING_FOR_DOWNLOAD	サーバがダウンロードの送信を開始するのを待機しています。
public final int	RECEIVING_TABLE	新しいテーブルがダウンロードされていることを示します。
public final int	RECEIVING_DATA	スキーマ情報またはローデータが受信されていることを示します。
public final int	COMMITTING_DOWNLOAD	ダウンロードされたローがデータベースにコミットされていることを示します。
public final int	ROLLING_BACK_DOWNLOAD	ダウンロードされたローがデータベースにコミットされていることを示します。
public final int	SENDING_DOWNLOAD_ACK	ダウンロード完了の確認が送信されていることを示します。
public final int	DISCONNECTING	同期ストリームを切断していることを示します。
public final int	DONE	同期を終了していることを示します。 その他のステータスはレポートされません。
public final int	ERROR	同期は完了しましたが、エラーが発生したことを示します。

関連情報

setSyncObserver(SyncObserver) メソッド [221 ページ]
SyncObserver インタフェース [194 ページ]

1.25 SyncParms クラス

データベース同期処理中に使用されたパラメータを保持します。

構文

```
public abstract class SyncParms
```

メンバー

SyncParms のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

変数

変更子とタイプ	変数	説明
public static final int	HTTP_STREAM	HTTP 同期用の SyncParms オブジェクトを作成します。
public static final int	HTTPS_STREAM	セキュア HTTPS 同期用の SyncParms オブジェクトを作成します。
public static final int	TCPIP_STREAM	TCP/IP 同期用の SyncParms オブジェクトを作成します。
public static final int	TLS_STREAM	TCP/IP 同期用の SyncParms オブジェクトを作成します。

メソッド

変更子とタイプ	メソッド	説明
public abstract boolean	getAcknowledgeDownload() [203 ページ]	クライアントがダウンロードの確認を送信しているかどうかを判断します。
public abstract String	getAdditionalParms() [203 ページ]	追加の同期パラメータを返します。
public abstract String	getAuthenticationParms() [204 ページ]	カスタムのユーザ認証スクリプトに渡されるパラメータを返します。
public abstract boolean	getKeepPartialDownload() [204 ページ]	部分ダウンロードが有効かどうかを判断します。
public abstract int	getLivenessTimeout() [205 ページ]	活性タイムアウトの長さを秒単位で返します。
public abstract String	getNewPassword() [205 ページ]	setUserName メソッドで指定されたユーザの新しい Mobile Link パスワードを返します。
public abstract String	getPassword() [206 ページ]	setUserName メソッドで指定されたユーザの Mobile Link パスワードを返します。
public abstract String	getPublications() [206 ページ]	同期させるパブリケーションを返します。

変更子とタイプ	メソッド	説明
public abstract boolean	getResumePartialDownload() [207 ページ]	部分ダウンロードを再開する必要があるかどうかを判断します。
public abstract StreamHTTPParms	getStreamParms() [207 ページ]	同期ストリームの設定に使用するパラメータを返します。
public abstract SyncObserver	getSyncObserver() [208 ページ]	現在指定されている SyncObserver オブジェクトを返します。
public abstract SyncResult	getSyncResult() [208 ページ]	同期のステータスを含む SyncResult オブジェクトを返します。
public abstract String	getTableOrder() [209 ページ]	統合データベースにテーブルがアップロードされる順序を返します。
public abstract String	getUserName() [210 ページ]	Mobile Link サーバがクライアントをユニークに識別する Mobile Link ユーザ名を返します。
public abstract String	getVersion() [210 ページ]	使用するスクリプトバージョンを返します。
public abstract boolean	isDownloadOnly() [211 ページ]	同期がダウンロード専用かどうかを確認します。
public abstract boolean	isPingOnly() [211 ページ]	クライアントが Mobile Link サーバに対して ping または同期を実行しているかどうかを確認します。
public abstract boolean	isUploadOnly() [212 ページ]	同期がアップロード専用かどうかを確認します。
public abstract void	setAcknowledgeDownload(boolean) [212 ページ]	クライアントがダウンロードの確認を送信すべきかどうかを指定します。
public abstract void	setAdditionalParms(String) [213 ページ]	"名前=値" のペアをセミコロンで区切ったリストで、追加の同期パラメータを指定します。
public abstract void	setAuthenticationParms(String) [214 ページ]	カスタムユーザ認証スクリプト (Mobile Link authenticate_parameters 接続イベント) のパラメータを指定します。
public abstract void	setDownloadOnly(boolean) [214 ページ]	同期をダウンロード専用に設定します。
public abstract void	setKeepPartialDownload(boolean) [215 ページ]	同期中に部分ダウンロードを許可するかどうかを指定します。
public abstract void	setLivenessTimeout(int) [216 ページ]	活性タイムアウトの長さを秒単位で設定します。
public abstract void	setNewPassword(String) [217 ページ]	setUserName メソッドで指定されたユーザの新しい Mobile Link パスワードを設定します。
public abstract void	setPassword(String) [218 ページ]	setUserName メソッドで指定されたユーザの Mobile Link パスワードを設定します。
public abstract void	setPingOnly(boolean) [218 ページ]	クライアントが Mobile Link サーバに対して、同期を実行しないで ping を実行するように設定します。
public abstract void	setPublications(String) [219 ページ]	同期させるパブリケーションを設定します。

変更子とタイプ	メソッド	説明
public abstract void	setResumePartialDownload(boolean) [220 ページ]	前の部分的なダウンロードを再開するか、破棄するかを指定します。
public abstract void	setSyncObserver(SyncObserver) [221 ページ]	同期の進行状況をモニタする SyncObserver オブジェクトを設定します。
public abstract void	setTableOrder(String) [221 ページ]	統合データベースにテーブルがアップロードされる順序を設定します。
public abstract void	setUploadOnly(boolean) [222 ページ]	同期をアップロード専用に設定します。
public abstract void	setUserName(String) [223 ページ]	Mobile Link サーバがクライアントをユニークに識別する Mobile Link ユーザ名を設定します。
public abstract void	setVersion(String) [224 ページ]	使用する同期スクリプトを設定します。

備考

このインタフェースは、Connection.createSyncParms メソッドによって呼び出されます。

同期コマンドは一度に 1 つだけ設定できます。コマンドは、setDownloadOnly、setPingOnly、setUploadOnly の各メソッドを使用して指定します。このいずれかのメソッドを true に設定すると、他のメソッドが false に設定されます。

UserName パラメータと Version パラメータは設定する必要があります。UserName はクライアントデータベースごとにユニークである必要があります。

通信ストリームは、getStreamParms メソッドを使用して、SyncParms オブジェクトのタイプに基づいて設定します。たとえば、次のコードでは、HTTP 同期の準備と実行を行っています。

```
SyncParms syncParms = myConnection.createSyncParms(
   SyncParms.HTTP_STREAM,
   "MyUniqueMLUserID",
   "MyMLScriptVersion"
);
syncParms.setPassword("ThePWDforMyUniqueMLUserID");
syncParms.getStreamParms().setHost("MyMLHost");
myConnection.synchronize(syncParms);
```

カンマ区切りのリスト

AuthenticationParms、Publications、TableOrder の各パラメータはすべて、値のカンマ区切りのリストを含む文字列値を使用して指定します。リスト内の値は一重引用符または二重引用符で囲むことができますが、エスケープ文字はありません。引用符がなかった場合、値の前後のスペースは無視されます。たとえば、次のコードは、*Table A*、*Table B,D*、*Table C* を順に設定します。

```
syncParms.setTableOrder( "'Table A',¥"Table B,D",Table C );
```

このセクションの内容:

getAcknowledgeDownload() メソッド [203 ページ]
　クライアントがダウンロードの確認を送信しているかどうかを判断します。

setDownloadOnly(boolean) メソッド [214 ページ]
同期をダウンロード専用に設定します。

setKeepPartialDownload(boolean) メソッド [215 ページ]
同期中に部分ダウンロードを許可するかどうかを指定します。

setLivenessTimeout(int) メソッド [216 ページ]
活性タイムアウトの長さを秒単位で設定します。

setNewPassword(String) メソッド [217 ページ]
setUserName メソッドで指定されたユーザの新しい Mobile Link パスワードを設定します。

setPassword(String) メソッド [218 ページ]
setUserName メソッドで指定されたユーザの Mobile Link パスワードを設定します。

setPingOnly(boolean) メソッド [218 ページ]
クライアントが Mobile Link サーバに対して、同期を実行しないで ping を実行するように設定します。

setPublications(String) メソッド [219 ページ]
同期させるパブリケーションを設定します。

setResumePartialDownload(boolean) メソッド [220 ページ]
前の部分的なダウンロードを再開するか、破棄するかを指定します。

setSyncObserver(SyncObserver) メソッド [221 ページ]
同期の進行状況をモニタする SyncObserver オブジェクトを設定します。

setTableOrder(String) メソッド [221 ページ]
統合データベースにテーブルがアップロードされる順序を設定します。

setUploadOnly(boolean) メソッド [222 ページ]
同期をアップロード専用に設定します。

setUserName(String) メソッド [223 ページ]
Mobile Link サーバがクライアントをユニークに識別する Mobile Link ユーザ名を設定します。

setVersion(String) メソッド [224 ページ]
使用する同期スクリプトを設定します。

関連情報

getStreamParms() メソッド [207 ページ]
setUserName(String) メソッド [223 ページ]
createSyncParms(int, String, String) メソッド [43 ページ]
StreamHTTPParms インタフェース [171 ページ]
StreamHTTPSParms インタフェース [184 ページ]

1.25.1 getAcknowledgeDownload() メソッド

クライアントがダウンロードの確認を送信しているかどうかを判断します。

構文

```
public abstract boolean getAcknowledgeDownload ()
```

戻り値

クライアントがダウンロードの確認を送信している場合は true、それ以外の場合は false。

関連情報

setAcknowledgeDownload(boolean) メソッド [212 ページ]

1.25.2 getAdditionalParms() メソッド

追加の同期パラメータを返します。

構文

```
public abstract String getAdditionalParms ()
```

戻り値

追加パラメータのリスト、またはパラメータが指定されていない場合は NULL。

関連情報

setAdditionalParms(String) メソッド [213 ページ]

1.25.3 getAuthenticationParms() メソッド

カスタムのユーザ認証スクリプトに渡されるパラメータを返します。

構文

```
public abstract String getAuthenticationParms ()
```

戻り値

認証パラメータのリスト、またはパラメータが指定されていない場合は NULL。

関連情報

setAuthenticationParms(String) メソッド [214 ページ]

1.25.4 getKeepPartialDownload() メソッド

部分ダウンロードが有効かどうかを判断します。

構文

```
public abstract boolean getKeepPartialDownload ()
```

戻り値

部分ダウンロードが有効になっている場合は true、それ以外の場合は false。

関連情報

setKeepPartialDownload(boolean) メソッド [215 ページ]

1.25.5 getLivenessTimeout() メソッド

活性タイムアウトの長さを秒単位で返します。

構文

```
public abstract int getLivenessTimeout ()
```

戻り値

タイムアウト。

関連情報

setLivenessTimeout(int) メソッド [216 ページ]

1.25.6 getNewPassword() メソッド

setUserName メソッドで指定されたユーザの新しい Mobile Link パスワードを返します。

構文

```
public abstract String getNewPassword ()
```

戻り値

次の同期後に設定される新しいパスワード。

関連情報

setUserName(String) メソッド [223 ページ]
setNewPassword(String) メソッド [217 ページ]

1.25.7 getPassword() メソッド

setUserName メソッドで指定されたユーザの Mobile Link パスワードを返します。

構文

```
public abstract String getPassword ()
```

戻り値

Mobile Link ユーザのパスワード。

関連情報

setPassword(String) メソッド [218 ページ]

1.25.8 getPublications() メソッド

同期させるパブリケーションを返します。

構文

```
public abstract String getPublications ()
```

戻り値

同期するパブリケーションのセット。

関連情報

setPublications(String) メソッド [219 ページ]

1.25.9 getResumePartialDownload() メソッド

部分ダウンロードを再開する必要があるかどうかを判断します。

構文

```
public abstract boolean getResumePartialDownload ()
```

戻り値

部分ダウンロードが再開される場合は true、それ以外の場合は false。

関連情報

setResumePartialDownload(boolean) メソッド [220 ページ]

1.25.10 getStreamParms() メソッド

同期ストリームの設定に使用するパラメータを返します。

構文

```
public abstract StreamHTTPParms getStreamParms ()
```

戻り値

HTTP または HTTPS の同期ストリームのパラメータを指定する StreamHTTPParms または StreamHTTPSParms オブジェクト。オブジェクトは参照で返されます。

備考

同期ストリームのタイプは、SyncParms オブジェクトの作成時に指定します。

関連情報

createSyncParms(int, String, String) メソッド [43 ページ]
StreamHTTPParms インタフェース [171 ページ]
StreamHTTPSParms インタフェース [184 ページ]

1.25.11 getSyncObserver() メソッド

現在指定されている SyncObserver オブジェクトを返します。

構文

```
public abstract SyncObserver getSyncObserver ()
```

戻り値

SyncObserver オブジェクト、または observer が指定されていない場合は NULL。

関連情報

setSyncObserver(SyncObserver) メソッド [221 ページ]

1.25.12 getSyncResult() メソッド

同期のステータスを含む SyncResult オブジェクトを返します。

構文

```
public abstract SyncResult getSyncResult ()
```

戻り値

前回の Connection.synchronize メソッドの呼び出しの結果を表す SyncResult オブジェクト。

備考

次の例は、前回の Connection.synchronize メソッド呼び出しの結果セットを取得する方法を示しています。

```
conn.synchronize( mySyncParms );
SyncResult result = mySyncParms.getSyncResult();
display(
   "*** Synchronized *** sent=" + result.getSentRowCount()
   + ", received=" + result.getReceivedRowCount()
);
```

注記

このメソッドは、前回の SYNCHRONIZE SQL 文の結果を返しません。前回の SYNCHRONIZE SQL 文の SyncResult オブジェクトを取得するには、渡された Connection オブジェクトで getSyncResult メソッドを使用します。

関連情報

SyncResult クラス [224 ページ]
getSyncResult() メソッド [49 ページ]

1.25.13 getTableOrder() メソッド

統合データベースにテーブルがアップロードされる順序を返します。

構文

```
public abstract String getTableOrder ()
```

戻り値

テーブル名のカンマ区切りのリスト、またはテーブルの順序が指定されていない場合は NULL。カンマ区切りのリストの詳細については、クラスの説明を参照してください。

関連情報

setTableOrder(String) メソッド [221 ページ]

1.25.14 getUserName() メソッド

Mobile Link サーバがクライアントをユニークに識別する Mobile Link ユーザ名を返します。

構文

```
public abstract String getUserName ()
```

戻り値

Mobile Link ユーザ名

関連情報

setUserName(String) メソッド [223 ページ]

1.25.15 getVersion() メソッド

使用するスクリプトバージョンを返します。

構文

```
public abstract String getVersion ()
```

戻り値

スクリプトバージョン。

関連情報

setVersion(String) メソッド [224 ページ]

1.25.16 isDownloadOnly() メソッド

同期がダウンロード専用かどうかを確認します。

構文

```
public abstract boolean isDownloadOnly ()
```

戻り値

アップロードが無効になっている場合は true、それ以外の場合は false。

関連情報

setDownloadOnly(boolean) メソッド [214 ページ]

1.25.17 isPingOnly() メソッド

クライアントが Mobile Link サーバに対して ping または同期を実行しているかどうかを確認します。

構文

```
public abstract boolean isPingOnly ()
```

戻り値

クライアントがサーバに対して ping だけを実行している場合は true、それ以外の場合は false。

関連情報

setPingOnly(boolean) メソッド [218 ページ]

1.25.18 isUploadOnly() メソッド

同期がアップロード専用かどうかを確認します。

構文

```
public abstract boolean isUploadOnly ()
```

戻り値

ダウンロードが無効になっている場合は true、それ以外の場合は false。

関連情報

setUploadOnly(boolean) メソッド [222 ページ]

1.25.19 setAcknowledgeDownload(boolean) メソッド

クライアントがダウンロードの確認を送信すべきかどうかを指定します。

構文

```
public abstract void setAcknowledgeDownload (boolean ack)
```

パラメータ

ack クライアントからダウンロード確認を送信する場合は true、それ以外の場合は false に設定します。

備考

デフォルトは false です。

関連情報

getAcknowledgeDownload() メソッド [203 ページ]

1.25.20 setAdditionalParms(String) メソッド

"名前=値" のペアをセミコロンで区切ったリストで、追加の同期パラメータを指定します。

構文

```
public abstract void setAdditionalParms (String v) throws ULjException
```

パラメータ

v "名前=値" のペアをセミコロンで区切ったリスト形式の文字列。

備考

このメソッドは、SyncParms クラスの既存のメソッドで指定できない同期パラメータをいくつか追加指定する場合に使用します。

次の例は、SyncParms オブジェクトに AllowDownloadDupRows、CheckpointStore、DisableConcurrency のパラメータを設定する方法を示します。

```
SyncParms parms;
...
parms.setAdditionalParms(
  "AllowDownloadDupRows=1;CheckpointStore=1;DisableConcurrency=1" );
```

関連情報

getAdditionalParms() メソッド [203 ページ]

1.25.21 setAuthenticationParms(String) メソッド

カスタムユーザ認証スクリプト (Mobile Link authenticate_parameters 接続イベント) のパラメータを指定します。

構文

```
public abstract void setAuthenticationParms (String v) throws ULjException
```

パラメータ

v 認証パラメータのカンマ区切りのリスト、または NULL を参照してください。カンマ区切りのリストの詳細については、クラスの説明を参照してください。

備考

最初の 255 文字列のみが使用されます。それぞれの文字列は認証パラメータの Mobile Link サーバの制限よりも長くはなりません (現在は 4000 UTF8 バイト)。

21K 文字よりも長い文字列は、Mobile Link に送信されたときにトランケートされ、認証パラメータ用のサーバ制限を超過する文字列はサーバ側の同期エラーを引き起こします。

関連情報

getAuthenticationParms() メソッド [204 ページ]

1.25.22 setDownloadOnly(boolean) メソッド

同期をダウンロード専用に設定します。

構文

```
public abstract void setDownloadOnly (boolean v)
```

パラメータ

v アップロードを無効にする場合は true、有効にする場合は false に設定します。

備考

デフォルトは false です。true を指定すると、setPingOnly メソッドと setUploadOnly メソッドが自動的に呼び出され、これらが false に設定されます。

関連情報

isDownloadOnly() メソッド [211 ページ]
setPingOnly(boolean) メソッド [218 ページ]
setUploadOnly(boolean) メソッド [222 ページ]

1.25.23 setKeepPartialDownload(boolean) メソッド

同期中に部分ダウンロードを許可するかどうかを指定します。

構文

```
public abstract void setKeepPartialDownload (boolean c) throws ULjException
```

パラメータ

c 部分ダウンロードを有効にする場合は、true に設定します。

備考

デフォルト設定は false です。同期中に部分ダウンロードを有効にして保存する場合は true に設定します。そうではなく、部分ダウンロードを無効にしてエラーが発生したときにダウンロードをロールバックする場合は false に設定します。

Ultra Light で、通信エラーや SyncObserver オブジェクトによるアボートのために失敗したダウンロードを、部分的に再開できるようになりました。Ultra Light は、ダウンロードを受信しながら処理します。ダウンロードが中断した場合は、部分的なダウンロードトランザクションがデータベース内に残るため、次の同期中に再開できます。

Ultra Light で部分的なダウンロードを保存する必要があることを示すには、true を指定します。指定しないと、エラーが発生した場合にダウンロードがロールバックされます。

部分ダウンロードが維持された場合、Connection.synchronize メソッドの終了時に SyncResult.getPartialDownloadRetained メソッドが true を返します。

KeepPartialDownload 同期パラメータが true に設定されていれば、部分ダウンロードを再開できます。部分ダウンロードを再開するには、setResumePartialDownload メソッドを true に設定して、Connection.synchronize メソッドを呼び出します。

別の通信エラーの発生に備えて、KeepPartialDownload 同期パラメータも true に設定しておきます。ダウンロードが省略された場合は、アップロードは行われません。

再開したダウンロードで受信するダウンロードは、最初にダウンロードを開始したときと同じものです。最新のデータが必要な場合は、再開されたダウンロードが完了した直後に、もう一度ダウンロードを行うことができます。

ダウンロードの再開時には、SyncParms クラスで指定される同期パラメータの多くは使用されません。たとえば、Publications パラメータは使用されません。受信するパブリケーションは、最初のダウンロード時に要求したものです。使用する必要があるのは、setResumePartialDownload メソッドと setUserName メソッドだけです。setKeepPartialDownload メソッドは必要に応じて使用できます。

部分的なダウンロードが存在するが、このダウンロードが必要ではなくなった場合は、Connection.rollbackPartialDownload を呼び出して、失敗したダウンロードトランザクションをロールバックできます。また、同期を再試行するときに ResumePartialDownload パラメータを指定しなかった場合は、次の同期が開始される前に、部分的なダウンロードがロールバックされます。

関連情報

getKeepPartialDownload() メソッド [204 ページ]
setResumePartialDownload(boolean) メソッド [220 ページ]
setUserName(String) メソッド [223 ページ]

1.25.24 setLivenessTimeout(int) メソッド

活性タイムアウトの長さを秒単位で設定します。

構文

```
public abstract void setLivenessTimeout (int seconds) throws ULjException
```

パラメータ

seconds 新しい活性タイムアウト値。

備考

活性タイムアウトは、サーバで許容される、リモートのアイドル時間の長さです。リモートが 1 秒間サーバと通信しなかった場合、サーバはリモートとの接続が失われたとみなし、同期を終了します。リモートは、接続を継続するために、自動的に定期メッセージをサーバに送信します。

負の値を設定すると、例外がスローされます。値は Mobile Link サーバによって予告なく変更される場合があります。変更は、値が低すぎるか高すぎる場合に行われます。

デフォルト値は 240 秒です。

関連情報

getLivenessTimeout() メソッド [205 ページ]

1.25.25 setNewPassword(String) メソッド

setUserName メソッドで指定されたユーザの新しい Mobile Link パスワードを設定します。

構文

```
public abstract void setNewPassword (String v)
```

パラメータ

v Mobile Link ユーザの新しいパスワード。

備考

新しいパスワードが有効になるのは、次の同期の後です。

デフォルトは NULL で、この場合、パスワードは置換されません。

関連情報

getNewPassword() メソッド [205 ページ]
setPassword(String) メソッド [218 ページ]

setUserName(String) メソッド [223 ページ]

1.25.26 setPassword(String) メソッド

setUserName メソッドで指定されたユーザの Mobile Link パスワードを設定します。

構文

```
public abstract void setPassword (String v) throws ULjException
```

パラメータ

v Mobile Link ユーザのパスワード。

備考

このユーザ名とパスワードは、データベースのユーザ ID とパスワードとは異なります。このメソッドは、Mobile Link サーバに対してアプリケーションを認証するために使用されます。

デフォルトは空の文字列で、これはパスワードなしを示します。

関連情報

getPassword() メソッド [206 ページ]
setNewPassword(String) メソッド [217 ページ]
setUserName(String) メソッド [223 ページ]

1.25.27 setPingOnly(boolean) メソッド

クライアントが Mobile Link サーバに対して、同期を実行しないで ping を実行するように設定します。

構文

```
public abstract void setPingOnly (boolean v)
```

パラメータ

v サーバに ping だけを実行する場合は true、同期を実行する場合は false に設定します。

備考

デフォルトは false です。true を指定すると、setDownloadOnly メソッドと setUploadOnly メソッドが自動的に呼び出され、これらが false に設定されます。

関連情報

isPingOnly() メソッド [211 ページ]
setDownloadOnly(boolean) メソッド [214 ページ]
setUploadOnly(boolean) メソッド [222 ページ]

1.25.28 setPublications(String) メソッド

同期させるパブリケーションを設定します。

構文

```
public abstract void setPublications (String pubs) throws ULjException
```

パラメータ

pubs パブリケーション名のカンマ区切りのリスト。カンマ区切りのリストの詳細については、クラスの説明を参照してください。

備考

デフォルトでは、データベース内のすべてのテーブルの同期を指定する Connection.SYNC_ALL 定数が設定されます。すべてのパブリケーションを同期するには、このメソッドに Connection.SYNC_ALL_PUBS 定数を設定します。

関連情報

getPublications() メソッド [206 ページ]

1.25.29 setResumePartialDownload(boolean) メソッド

前の部分的なダウンロードを再開するか、破棄するかを指定します。

構文

```
public abstract void setResumePartialDownload (boolean c) throws ULjException
```

パラメータ

c 以前の部分ダウンロードを再開する場合は true に設定します。

例外

ULjException class 次の同期パラメータ (DownloadOnly、PingOnly、ResumePartialDownload、UploadOnly) のうち 1 つ以上が true に設定されると、Connection.synchronize メソッドによって SQLE_SYNC_INFO_INVALID がスローされます。

備考

前回の部分的なダウンロードを再開する場合は true、破棄する場合は false に設定します。デフォルト設定は false です。

関連情報

getResumePartialDownload() メソッド [207 ページ]

1.25.30 setSyncObserver(SyncObserver) メソッド

同期の進行状況をモニタする SyncObserver オブジェクトを設定します。

構文

```
public abstract void setSyncObserver (SyncObserver so)
```

パラメータ

so SyncObserver オブジェクト。

備考

デフォルトは NULL で、これは observer なしを示します。

関連情報

SyncObserver インタフェース [194 ページ]

1.25.31 setTableOrder(String) メソッド

統合データベースにテーブルがアップロードされる順序を設定します。

構文

```
public abstract void setTableOrder (String v) throws ULjException
```

パラメータ

v 同期する順序でのテーブル名のカンマ区切りのリスト、またはテーブル順序を指定しない場合は NULL。カンマ区切りのリストの詳細については、クラスの説明を参照してください。

備考

プライマリテーブルをリストの先頭に指定し、統合データベースで外部キー関係を持つすべてのテーブルをリストに含めます。

Publications パラメータによって同期対象として選択されているテーブルはすべて、TableOrder パラメータで指定されているかどうかに関係なく同期されます。指定されていないテーブルは、クライアントデータベースでの外部キー関係の順序で同期されます。これらは、指定したテーブルの後に同期されます。

デフォルト値は NULL 参照で、テーブルのデフォルトの順序は上書きされません。

関連情報

getTableOrder() メソッド [209 ページ]
setPublications(String) メソッド [219 ページ]

1.25.32 setUploadOnly(boolean) メソッド

同期をアップロード専用に設定します。

構文

```
public abstract void setUploadOnly (boolean v)
```

パラメータ

v ダウンロードを無効にする場合は true、有効にする場合は false に設定します。

備考

デフォルトは false です。true を指定すると、setDownloadOnly メソッドと setPingOnly メソッドが自動的に呼び出され、これらが false に設定されます。

関連情報

isUploadOnly() メソッド [212 ページ]
setDownloadOnly(boolean) メソッド [214 ページ]

setPingOnly(boolean) メソッド [218 ページ]

1.25.33 setUserName(String) メソッド

Mobile Link サーバがクライアントをユニークに識別する Mobile Link ユーザ名を設定します。

構文

```
public abstract void setUserName (String v) throws ULjException
```

パラメータ

v Mobile Link ユーザ名

備考

この値は、以下を判別するために使用されます。

- ダウンロードの内容
- 同期ステータスを記録するかどうか
- 同期中に中断された場合、回復するかどうか

このユーザ名とパスワードは、データベースのユーザ ID とパスワードとは異なります。このメソッドは、Mobile Link サーバに対してアプリケーションを認証するために使用されます。

このパラメータは、SyncParms オブジェクトの作成時に初期化されます。

関連情報

getUserName() メソッド [210 ページ]
setPassword(String) メソッド [218 ページ]
setNewPassword(String) メソッド [217 ページ]
createSyncParms(int, String, String) メソッド [43 ページ]

1.25.34 setVersion(String) メソッド

使用する同期スクリプトを設定します。

構文

```
public abstract void setVersion (String v) throws ULjException
```

パラメータ

v スクリプトバージョン。

備考

統合データベースの同期スクリプトは、それぞれバージョン文字列でマーク付けされます。たとえば、異なるバージョン文字列によって特定される 2 つの download_cursor スクリプトが存在する場合があります。バージョン文字列によって、アプリケーションが同期スクリプトのセットから適切に選択できます。

このパラメータは、SyncParms オブジェクトの作成時に初期化されます。

関連情報

getVersion() メソッド [210 ページ]
createSyncParms(int, String, String) メソッド [43 ページ]

1.26 SyncResult クラス

指定されたデータベース同期のステータス関連の情報をレポートします。

構文

```
public abstract class SyncResult
```

メンバー

SyncResult のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public abstract String	getAuthMessage() [227 ページ]	カスタムユーザ認証同期スクリプトで指定されている、前回の同期試行の承認メッセージを返します。
public abstract int	getAuthStatus() [228 ページ]	前回試行された同期の認証ステータスコードを返します。
public abstract int	getAuthValue() [228 ページ]	カスタムユーザ認証同期スクリプトで指定されている値を返します。
public abstract String	getCurrentTableName() [229 ページ]	現在同期中のテーブルの名前を返します。
public abstract boolean	getIgnoredRows() [229 ページ]	前回行われた同期で、アップロードされたローが無視されたかどうかを確認します。
public abstract boolean	getPartialDownloadRetained() [229 ページ]	前回行われた同期で、部分的なダウンロードが保持されたかどうかを確認します。
public abstract long	getReceivedByteCount() [230 ページ]	データ同期中に受信したバイト数を返します。
public abstract long	getReceivedDeletes() [230 ページ]	受信したローのうち削除されたものの数を返します。
public abstract long	getReceivedIgnoredDeletes() [231 ページ]	受信した削除ローのうち無視されたものの数を返します。
public abstract long	getReceivedIgnoredUpdates() [231 ページ]	受信した更新ローのうち無視されたものの数を返します。
public abstract long	getReceivedInserts() [232 ページ]	受信したローのうち挿入されたものの数を返します。
public abstract long	getReceivedRowCount() [233 ページ]	受信したローの数を返します。
public abstract long	getReceivedTruncateDeletes() [233 ページ]	受信したローのうち、ダウンロードされたトランケート操作によって削除されたものの数を返します。
public abstract long	getReceivedUpdates() [234 ページ]	受信したローのうち更新として適用されたものの数を返します。
public abstract long	getSentByteCount() [235 ページ]	データ同期中に送信されたバイト数を返します。
public abstract long	getSentDeletes() [235 ページ]	送信された削除済みローの数を返します。
public abstract long	getSentInserts() [236 ページ]	送信された挿入済みローの数を返します。
public abstract long	getSentUpdates() [236 ページ]	送信された更新済みローの数を返します。
public abstract int	getStreamErrorCode() [237 ページ]	ストリーム自体によってレポートされるエラーコードを返します。

変更子とタイプ	メソッド	説明
public abstract String	getStreamErrorMessage() [237 ページ]	ストリーム自体によってレポートされるエラーメッセージを返します。
public abstract int	getSyncedTableCount() [238 ページ]	現在までに同期されたテーブル数を返します。
public abstract long	getTotalDownloadRowCount() [238 ページ]	ダウンロードで受信するローの合計数を返します。
public abstract int	getTotalTableCount() [239 ページ]	同期されるテーブル数を返します。
public abstract boolean	isUploadOK() [239 ページ]	前回のアップロード同期が成功したかどうかを確認します。

このセクションの内容:

getAuthMessage() メソッド [227 ページ]
カスタムユーザ認証同期スクリプトで指定されている、前回の同期試行の承認メッセージを返します。

getAuthStatus() メソッド [228 ページ]
前回試行された同期の認証ステータスコードを返します。

getAuthValue() メソッド [228 ページ]
カスタムユーザ認証同期スクリプトで指定されている値を返します。

getCurrentTableName() メソッド [229 ページ]
現在同期中のテーブルの名前を返します。

getIgnoredRows() メソッド [229 ページ]
前回行われた同期で、アップロードされたローが無視されたかどうかを確認します。

getPartialDownloadRetained() メソッド [229 ページ]
前回行われた同期で、部分的なダウンロードが保持されたかどうかを確認します。

getReceivedByteCount() メソッド [230 ページ]
データ同期中に受信したバイト数を返します。

getReceivedDeletes() メソッド [230 ページ]
受信したローのうち削除されたものの数を返します。

getReceivedIgnoredDeletes() メソッド [231 ページ]
受信した削除ローのうち無視されたものの数を返します。

getReceivedIgnoredUpdates() メソッド [231 ページ]
受信した更新ローのうち無視されたものの数を返します。

getReceivedInserts() メソッド [232 ページ]
受信したローのうち挿入されたものの数を返します。

getReceivedRowCount() メソッド [233 ページ]
受信したローの数を返します。

getReceivedTruncateDeletes() メソッド [233 ページ]
受信したローのうち、ダウンロードされたトランケート操作によって削除されたものの数を返します。

getReceivedUpdates() メソッド [234 ページ]
受信したローのうち更新として適用されたものの数を返します。

getSentByteCount() メソッド [235 ページ]
　データ同期中に送信されたバイト数を返します。

getSentDeletes() メソッド [235 ページ]
　送信された削除済みローの数を返します。

getSentInserts() メソッド [236 ページ]
　送信された挿入済みローの数を返します。

getSentUpdates() メソッド [236 ページ]
　送信された更新済みローの数を返します。

getStreamErrorCode() メソッド [237 ページ]
　ストリーム自体によってレポートされるエラーコードを返します。

getStreamErrorMessage() メソッド [237 ページ]
　ストリーム自体によってレポートされるエラーメッセージを返します。

getSyncedTableCount() メソッド [238 ページ]
　現在までに同期されたテーブル数を返します。

getTotalDownloadRowCount() メソッド [238 ページ]
　ダウンロードで受信するローの合計数を返します。

getTotalTableCount() メソッド [239 ページ]
　同期されるテーブル数を返します。

isUploadOK() メソッド [239 ページ]
　前回のアップロード同期が成功したかどうかを確認します。

関連情報

getSyncResult() メソッド [208 ページ]

1.26.1 getAuthMessage() メソッド

カスタムユーザ認証同期スクリプトで指定されている、前回の同期試行の承認メッセージを返します。

構文

```
public abstract String getAuthMessage ()
```

戻り値

最後の同期の認証に関する情報が含まれた文字列。

備考

ブランクまたは空のメッセージは NULL として返されます。

認証ステータスコードとして認証メッセージが返される場合もあります。詳細については、Mobile Link 名前付きシステムパラメータ authentication_message を参照してください。

1.26.2 getAuthStatus() メソッド

前回試行された同期の認証ステータスコードを返します。

構文

```
public abstract int getAuthStatus ()
```

戻り値

AuthStatusCode の値。

1.26.3 getAuthValue() メソッド

カスタムユーザ認証同期スクリプトで指定されている値を返します。

構文

```
public abstract int getAuthValue ()
```

戻り値

カスタムユーザ認証同期スクリプトから返された整数。

1.26.4 getCurrentTableName() メソッド

現在同期中のテーブルの名前を返します。

構文

```
public abstract String getCurrentTableName ()
```

戻り値

テーブル名。

1.26.5 getIgnoredRows() メソッド

前回行われた同期で、アップロードされたローが無視されたかどうかを確認します。

構文

```
public abstract boolean getIgnoredRows ()
```

戻り値

前回の同期中にアップロードされたローが無視された場合は true、ローが無視されなかった場合は false。

1.26.6 getPartialDownloadRetained() メソッド

前回行われた同期で、部分的なダウンロードが保持されたかどうかを確認します。

構文

```
public abstract boolean getPartialDownloadRetained ()
```

戻り値

ダウンロードが中断され、部分的なダウンロードが保持された場合は true、ダウンロードが中断されなかった場合または部分的なダウンロードがロールバックされた場合は false。

1.26.7 getReceivedByteCount() メソッド

データ同期中に受信したバイト数を返します。

構文

```
public abstract long getReceivedByteCount ()
```

戻り値

バイト数。

1.26.8 getReceivedDeletes() メソッド

受信したローのうち削除されたものの数を返します。

構文

```
public abstract long getReceivedDeletes ()
```

戻り値

削除が適用されたダウンロード済みローの数。

関連情報

getReceivedIgnoredDeletes() メソッド [231 ページ]
getReceivedIgnoredUpdates() メソッド [231 ページ]
getReceivedInserts() メソッド [232 ページ]

getReceivedTruncateDeletes() メソッド [233 ページ]
getReceivedUpdates() メソッド [234 ページ]

1.26.9 getReceivedIgnoredDeletes() メソッド

受信した削除ローのうち無視されたものの数を返します。

構文

```
public abstract long getReceivedIgnoredDeletes ()
```

戻り値

無視されたダウンロード済み削除ローの数。

関連情報

getReceivedIgnoredUpdates() メソッド [231 ページ]
getReceivedDeletes() メソッド [230 ページ]
getReceivedInserts() メソッド [232 ページ]
getReceivedTruncateDeletes() メソッド [233 ページ]
getReceivedUpdates() メソッド [234 ページ]

1.26.10 getReceivedIgnoredUpdates() メソッド

受信した更新ローのうち無視されたものの数を返します。

構文

```
public abstract long getReceivedIgnoredUpdates ()
```

戻り値

無視されたダウンロード済み更新ローの数。

備考

受信された更新ローが無視されるのは、ダウンロードで重複するプライマリキーが許可されている場合だけですこれ以外の場合、ダウンロード済みの更新が重複すると、同期が失敗します。

関連情報

getReceivedIgnoredDeletes() メソッド [231 ページ]
getReceivedDeletes() メソッド [230 ページ]
getReceivedInserts() メソッド [232 ページ]
getReceivedTruncateDeletes() メソッド [233 ページ]
getReceivedUpdates() メソッド [234 ページ]

1.26.11 getReceivedInserts() メソッド

受信したローのうち挿入されたものの数を返します。

構文

```
public abstract long getReceivedInserts ()
```

戻り値

挿入として適用されたローの数。

関連情報

getReceivedDeletes() メソッド [230 ページ]
getReceivedIgnoredDeletes() メソッド [231 ページ]
getReceivedTruncateDeletes() メソッド [233 ページ]
getReceivedUpdates() メソッド [234 ページ]

1.26.12 getReceivedRowCount() メソッド

受信したローの数を返します。

構文

```
public abstract long getReceivedRowCount ()
```

戻り値

受け取ったローの数。

備考

この数には、ダウンロード適用時に無視される可能性のあるローが含まれています。

関連情報

getTotalDownloadRowCount() メソッド [238 ページ]

1.26.13 getReceivedTruncateDeletes() メソッド

受信したローのうち、ダウンロードされたトランケート操作によって削除されたものの数を返します。

構文

```
public abstract long getReceivedTruncateDeletes ()
```

戻り値

トランケートされたローの数。

備考

各ダウンロードトランケート操作は、getReceivedRowCount メソッドでカウントされたダウンロードロー内の 1 つのローのように見えますが、ゼロまたは多くのローがトランケートされる可能性があり、このメソッドでカウントされます。

関連情報

getReceivedDeletes() メソッド [230 ページ]
getReceivedIgnoredDeletes() メソッド [231 ページ]
getReceivedIgnoredUpdates() メソッド [231 ページ]
getReceivedInserts() メソッド [232 ページ]
getReceivedUpdates() メソッド [234 ページ]

1.26.14 getReceivedUpdates() メソッド

受信したローのうち更新として適用されたものの数を返します。

構文

```
public abstract long getReceivedUpdates ()
```

戻り値

更新として適用されたローの数。

関連情報

getReceivedDeletes() メソッド [230 ページ]
getReceivedIgnoredDeletes() メソッド [231 ページ]
getReceivedIgnoredUpdates() メソッド [231 ページ]
getReceivedInserts() メソッド [232 ページ]
getReceivedTruncateDeletes() メソッド [233 ページ]

1.26.15 getSentByteCount() メソッド

データ同期中に送信されたバイト数を返します。

構文

```
public abstract long getSentByteCount ()
```

戻り値

送信されたバイト数。

1.26.16 getSentDeletes() メソッド

送信された削除済みローの数を返します。

構文

```
public abstract long getSentDeletes ()
```

戻り値

送信された削除済みローの数。

備考

挿入、更新、削除の数は、同期しているテーブルに対して実行された操作の数とは異なる場合があります。これは、対象のローに対する操作はすべて 1 つに結合されるからです。

関連情報

getSentInserts() メソッド [236 ページ]
getSentUpdates() メソッド [236 ページ]

1.26.17 getSentInserts() メソッド

送信された挿入済みローの数を返します。

構文

```
public abstract long getSentInserts ()
```

戻り値

送信された挿入済みローの数。

備考

挿入、更新、削除の数は、同期しているテーブルに対して実行された操作の数とは異なる場合があります。これは、対象のローに対する操作はすべて 1 つに結合されるからです。

関連情報

getSentDeletes() メソッド [235 ページ]
getSentUpdates() メソッド [236 ページ]

1.26.18 getSentUpdates() メソッド

送信された更新済みローの数を返します。

構文

```
public abstract long getSentUpdates ()
```

戻り値

送信された更新済みローの数。

備考

挿入、更新、削除の数は、同期しているテーブルに対して実行された操作の数とは異なる場合があります。これは、対象のローに対する操作はすべて 1 つに結合されるからです。

関連情報

getSentDeletes() メソッド [235 ページ]
getSentInserts() メソッド [236 ページ]

1.26.19 getStreamErrorCode() メソッド

ストリーム自体によってレポートされるエラーコードを返します。

構文

```
public abstract int getStreamErrorCode ()
```

戻り値

通信ストリームエラーがなかった場合は 0、それ以外の場合はサーバからの応答コードを返します。

備考

このメソッドは、HTTP 応答コードを返します。

1.26.20 getStreamErrorMessage() メソッド

ストリーム自体によってレポートされるエラーメッセージを返します。

構文

```
public abstract String getStreamErrorMessage ()
```

戻り値

メッセージがない場合は NULL、それ以外の場合は応答メッセージを返します。

備考

このメソッドは、HTTP 応答メッセージを返します。

1.26.21 getSyncedTableCount() メソッド

現在までに同期されたテーブル数を返します。

構文

```
public abstract int getSyncedTableCount ()
```

戻り値

同期されたテーブル数。

1.26.22 getTotalDownloadRowCount() メソッド

ダウンロードで受信するローの合計数を返します。

構文

```
public abstract long getTotalDownloadRowCount ()
```

戻り値

ダウンロードで受信するローの数。この数には、適用されないローが含まれます (クライアント上にないローの削除など)。

備考

この数には、重複していて無視されるローも含まれます。この値は、同期によって最初のテーブルの SyncObserver.State.RECEIVING_TABLE ステータスが入力されるまで、設定されません。

1.26.23 getTotalTableCount() メソッド

同期されるテーブル数を返します。

構文

```
public abstract int getTotalTableCount ()
```

戻り値

同期するテーブル数。

1.26.24 isUploadOK() メソッド

前回のアップロード同期が成功したかどうかを確認します。

構文

```
public abstract boolean isUploadOK ()
```

戻り値

前回のアップロード同期が成功した場合は true、それ以外の場合は false。

1.27 SyncResult.AuthStatusCode インタフェース

Mobile Link サーバから返された認証コードを列挙します。

構文

```
public interface AuthStatusCode
```

メンバー

AuthStatusCode のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

変数

変更子とタイプ	変数	説明
public final int	UNKNOWN	認証ステータスが不明です。 このコードは、同期が実行されていないことを示します。
public final int	VALID	ユーザ ID とパスワードは、同期時には有効でした。
public final int	VALID_BUT_EXPIRES_SOON	ユーザ ID とパスワードは、同期時には有効でしたが、まもなく有効期限が切れます。
public final int	EXPIRED	ユーザ ID またはパスワードの有効期限が切れています。 認証に失敗しました。
public final int	INVALID	ユーザ ID またはパスワードが不正です。 認証に失敗しました。
public final int	IN_USE	ユーザ ID がすでに使用されています。 認証に失敗しました。

関連情報

getAuthStatus() メソッド [228 ページ]

1.28 TableSchema インタフェース

テーブルのスキーマを指定し、システムテーブルの名前を定義する定数を提供します。

構文

```
public interface TableSchema
```

メンバー

TableSchema のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

変数

変更子とタイプ	変数	説明
public static final String	SYS_TABLES	データベース内のテーブルに関する情報を含むシステムテーブルの名前です。
public static final String	SYS_COLUMNS	データベース内のテーブルカラムに関する情報を含むシステムテーブルの名前です。
public static final String	SYS_INDEXES	データベース内のテーブルインデックスに関する情報を含むシステムテーブルの名前です。
public static final String	SYS_INDEX_COLUMNS	データベース内のインデックスカラムに関する情報を含むシステムテーブルの名前です。
public static final String	SYS_PUBLICATIONS	データベースパブリケーションに関する情報を含むシステムテーブルの名前です。
public static final String	SYS_ARTICLES	パブリケーションのアーティクルに関する情報を含むシステムテーブルの名前です。
public static final String	SYS_INTERNAL	内部情報を含むシステムテーブルの名前です。
public static final String	SYS_FOREIGN_KEYS	データベース内の外部キーに関する情報を含むシステムテーブルの名前です。
public static final String	SYS_FKEY_COLUMNS	外部キーカラムに関する情報を含むシステムテーブルの名前です。
public static final String	SYS_ULDATA	システムの値に関する情報を含むシステムテーブルの名前です。
public static final String	SYS_ULDATA_INTERNAL	内部システムデータの型です。
public static final String	SYS_ULDATA_OPTION	オプションのシステムデータの型です。
public static final String	SYS_ULDATA_PROPERTY	プロパティシステムデータの型です。
public static final String	SYS_PRIMARY_INDEX	システムテーブルのプライマリキーインデックスの名前です。

変更子とタイプ	変数	説明
public static final short	TABLE_IS_SYSTEM	テーブルがシステムテーブルであることを示します。 この値は、SYS_TABLES テーブルの table_flags カラムで他のフラグと論理的に組み合わせることができます。
public static final short	TABLE_IS_NOSYNC	テーブルが非同期テーブルであることを示します。 この値は、TABLE_IS_DOWNLOAD_ONLY フラグを除き、SYS_TABLES テーブルの table_flags カラムで他のフラグと論理的に組み合わせることができます。
public static final short	TABLE_IS_DOWNLOAD_ONLY	テーブルがダウンロード専用テーブル (同期されたときにアップロードされるテーブル) であることを示します。 この値は、TABLE_IS_NOSYNC フラグを除き、SYS_TABLES テーブルの table_flags カラムで他のフラグと論理的に組み合わせることができます。

備考

このインタフェースにはテーブル関連の定数だけが含まれます。これには、システムテーブル名、テーブルフラグ、*sysuldata* システムテーブルのデータ型などがあります。

1.29 ULjEvent インタフェース

Ultra Light J API システムイベントを示します。

構文

```
public interface ULjEvent
```

メンバー

ULjEvent のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

変数

変更子とタイプ	変数	説明
public final short	TABLE_MODIFIED_EVENT	"テーブル変更済み" イベントタイプを示します。
public final short	COMMIT_EVENT	"コミット" イベントタイプを示します。
public final short	SYNC_COMPLETE_EVENT	"同期の完了" イベントタイプを示します。

メソッド

変更子とタイプ	メソッド	説明
public String	getParameter(String) [243 ページ]	イベントの名前付きパラメータを返します。
public short	getType() [243 ページ]	イベントタイプを返します。

このセクションの内容:

1.29.1 getParameter(String) メソッド

イベントの名前付きパラメータを返します。

構文

```
public String getParameter (String name) throws ULjException
```

パラメータ

name 値を取得するイベントの名前

1.29.2 getType() メソッド

イベントタイプを返します。

構文

```
public short getType ()
```

1.30 ULjException クラス

データベースからスローされた例外に取って代わります。

構文

```
public abstract class ULjException
```

メンバー

ULjException のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

コンストラクタ

変更子とタイプ	コンストラクタ	説明
protected	ULjException(String) [245 ページ]	新しい ULjException クラスを構築します。

メソッド

変更子とタイプ	メソッド	説明
public abstract ULjException	getCausingException() [245 ページ]	例外の原因となっている ULjException オブジェクトを返します。
public abstract int	getErrorCode() [245 ページ]	この例外に関連付けられているエラーコードを返します。
public abstract String	getParameter(short) [246 ページ]	指定されたエラーパラメータを返します。
public abstract short	getParameterCount() [246 ページ]	エラーパラメータの数を返します。
public abstract int	getSqlOffset() [247 ページ]	SQL 文字列内のエラーオフセットを返します。

このセクションの内容:

ULjException(String) コンストラクタ [245 ページ]
新しい ULjException クラスを構築します。

getCausingException() メソッド [245 ページ]
例外の原因となっている ULjException オブジェクトを返します。

getErrorCode() メソッド [245 ページ]
この例外に関連付けられているエラーコードを返します。

getParameter(short) メソッド [246 ページ]
指定されたエラーパラメータを返します。

getParameterCount() メソッド [246 ページ]
エラーパラメータの数を返します。

getSqlOffset() メソッド [247 ページ]

SQL 文字列内のエラーオフセットを返します。

1.30.1 ULjException(String) コンストラクタ

新しい ULjException クラスを構築します。

構文

```
protected ULjException (String text)
```

パラメータ

text メッセージテキストです。

1.30.2 getCausingException() メソッド

例外の原因となっている ULjException オブジェクトを返します。

構文

```
public abstract ULjException getCausingException ()
```

戻り値

原因となっている例外がない場合は NULL、それ以外の場合は ULjException オブジェクトを返します。

1.30.3 getErrorCode() メソッド

この例外に関連付けられているエラーコードを返します。

構文

```
public abstract int getErrorCode ()
```

戻り値

エラーコード。

1.30.4 getParameter(short) メソッド

指定されたエラーパラメータを返します。

構文

```
public abstract String getParameter (short param_no)
```

パラメータ

param_no 1 から始まるパラメータ番号。

戻り値

エラーパラメータ。

1.30.5 getParameterCount() メソッド

エラーパラメータの数を返します。

構文

```
public abstract short getParameterCount ()
```

戻り値

エラーパラメータ数。

1.30.6 getSqlOffset() メソッド

SQL 文字列内のエラーオフセットを返します。

構文

```
public abstract int getSqlOffset ()
```

戻り値

エラーメッセージに関連付けられている SQL 文字列がない場合は -1、それ以外の場合は、文字列内でエラーが発生した箇所の 0 から始まるオフセットを返します。

1.31 Unsigned64 クラス

符号なし 64 ビットのバイナリ値を実装します。

構文

```
public class Unsigned64
```

メンバー

Unsigned64 のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public static final long	add(long, long) [248 ページ]	2 つの値を加算し、結果をそれ自体に配置します。
public static final byte	compare [249 ページ]	2 つの整数値を比較します。
public static final long	divide(long, long) [250 ページ]	2 つの値を除算し、結果をそれ自体に配置します。
public static final long	multiply(long, long) [251 ページ]	2 つの値を乗算し、結果をそれ自体に配置します。
public static final long	remainder [252 ページ]	1 つの値を別の値で除算したときの余りを返します。

変更子とタイプ	メソッド	説明
public static final long	subtract(long, long) [253 ページ]	2 つの値を減算し、結果をそれ自体に配置します。

備考

このクラスの目的は、値を long 型整数として保持し、これらの値をこのクラスの静的メソッドを使用して解釈することです。

このクラスはインスタンスを作成できません。

このセクションの内容:

add(long, long) メソッド [248 ページ]
2 つの値を加算し、結果をそれ自体に配置します。

compare メソッド [249 ページ]
2 つの整数値を比較します。

divide(long, long) メソッド [250 ページ]
2 つの値を除算し、結果をそれ自体に配置します。

multiply(long, long) メソッド [251 ページ]
2 つの値を乗算し、結果をそれ自体に配置します。

remainder メソッド [252 ページ]
1 つの値を別の値で除算したときの余りを返します。

subtract(long, long) メソッド [253 ページ]
2 つの値を減算し、結果をそれ自体に配置します。

1.31.1 add(long, long) メソッド

2 つの値を加算し、結果をそれ自体に配置します。

構文

```
public static final long add (
    long v1,
    long v2
)
```

パラメータ

v1 最初のオペランド。

v2 2 番目のオペランド。

戻り値

2 つのオペランドの和。

1.31.2 compare メソッド

2 つの整数値を比較します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public static final byte	compare(int, int) [249 ページ]	2 つの整数値を比較します。
public static final byte	compare(long, long) [250 ページ]	2 つの long 値を比較します。

このセクションの内容:

compare(int, int) メソッド [249 ページ]
2 つの整数値を比較します。

compare(long, long) メソッド [250 ページ]
2 つの long 値を比較します。

1.31.2.1 compare(int, int) メソッド

2 つの整数値を比較します。

構文

```
public static final byte compare (
    int v1,
    int v2
)
```

パラメータ

v1 比較する最初の値。
v2 比較する 2 番目の値。

戻り値

v2 が v1 より大きい場合は -1、v1 と v2 が等しい場合は 0、v2 が v1 より小さい場合は 1。

1.31.2.2 compare(long, long) メソッド

2 つの long 値を比較します。

構文

```
public static final byte compare (
    long v1,
    long v2
)
```

パラメータ

v1 比較する最初の値。
v2 比較する 2 番目の値。

戻り値

v2 が v1 より大きい場合は -1、v1 と v2 が等しい場合は 0、v2 が v1 より小さい場合は 1。

1.31.3 divide(long, long) メソッド

2 つの値を除算し、結果をそれ自体に配置します。

構文

```
public static final long divide (
```

```
    long v1,
    long v2
)
```

パラメータ

v1 最初のオペランド。
v2 2 番目のオペランド。

戻り値

最初のオペランドを 2 番目のオペランドで割った値。

1.31.4 multiply(long, long) メソッド

2 つの値を乗算し、結果をそれ自体に配置します。

構文

```
public static final long multiply (
    long v1,
    long v2
)
```

パラメータ

v1 最初のオペランド。
v2 2 番目のオペランド。

戻り値

v1 と v2 の積。

1.31.5 remainder メソッド

1 つの値を別の値で除算したときの余りを返します。

オーバロードリスト

変更子とタイプ	オーバーロード名	説明
public static final long	remainder(long, long) [252 ページ]	1 つの値を別の値で除算したときの余りを返します。
public static final long	remainder(long, long, long) [253 ページ]	指定の指数で乗算した値を別の値から差し引いた余り (v1 - quot * v2) を返します。

このセクションの内容:

1.31.5.1 remainder(long, long) メソッド

1 つの値を別の値で除算したときの余りを返します。

構文

```
public static final long remainder (
    long v1,
    long v2
)
```

パラメータ

v1 被除数。

v2 除数。

戻り値

long 型整数として表される余り。

1.31.5.2 remainder(long, long, long) メソッド

指定の指数で乗算した値を別の値から差し引いた余り (v1 - quot * v2) を返します。

構文

```
public static final long remainder (
    long v1,
    long v2,
    long quot
)
```

パラメータ

v1 被除数。

v2 除数。

quot 商の値。

戻り値

long 型整数として表される余り。

1.31.6 subtract(long, long) メソッド

2 つの値を減算し、結果をそれ自体に配置します。

構文

```
public static final long subtract (
    long v1,
    long v2
)
```

パラメータ

v1 最初のオペランド。
v2 2 番目のオペランド。

戻り値

v1 から v2 を差し引いた値。

1.32 UUIDValue インタフェース

ユニーク識別子 (UUID またはユニバーサルユニーク識別子) オブジェクトを記述します。

構文

```
public interface UUIDValue
```

メンバー

UUIDValue のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public String	getString() [255 ページ]	UUIDValue オブジェクトの String 表現を返します。
public boolean	isNull() [256 ページ]	UUIDValue オブジェクトが NULL かどうかを確認します。
public void	set(String) [256 ページ]	UUIDValue オブジェクトに String 値を設定します。
public void	setNull() [256 ページ]	UUIDValue オブジェクトを NULL に設定します。

備考

このようなエンティティは、ユニーク識別子が必要であり任意の値を指定できる場合に便利です。

テーブルのカラムに値が指定されておらず、DEFAULT NEWID() 句を使用してカラムが作成された場合、UUIDValue は SQL INSERT 文でも作成できます。

createUUIDValue メソッドを使用して UUIDValue を作成する場合、Connection オブジェクトを使用できます。

このセクションの内容:

関連情報

createUUIDValue() メソッド [44 ページ]

1.32.1 getString() メソッド

UUIDValue オブジェクトの String 表現を返します。

構文

```
public String getString () throws ULjException
```

戻り値

String 値。

1.32.2 isNull() メソッド

UUIDValue オブジェクトが NULL かどうかを確認します。

構文

```
public boolean isNull ()
```

戻り値

オブジェクトが NULL の場合は true、NULL 以外の場合は false。

1.32.3 set(String) メソッド

UUIDValue オブジェクトに String 値を設定します。

構文

```
public void set (String value) throws ULjException
```

パラメータ

value String として表した数値。

1.32.4 setNull() メソッド

UUIDValue オブジェクトを NULL に設定します。

構文

```
public void setNull () throws ULjException
```

1.33 ValidateDatabaseProgressData インタフェース

ValidateDatabase プログレスデータをレポートします。

構文

```
public interface ValidateDatabaseProgressData
```

メンバー

ValidateDatabaseProgressData のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public abstract String[]	getParms() [257 ページ]	ステータス ID に関連付けられたパラメータ配列を返します。
public abstract short	getStatusId() [258 ページ]	検証操作のステータス ID を返します。

このセクションの内容:

getParms() メソッド [257 ページ]
ステータス ID に関連付けられたパラメータ配列を返します。

getStatusId() メソッド [258 ページ]
検証操作のステータス ID を返します。

1.33.1 getParms() メソッド

ステータス ID に関連付けられたパラメータ配列を返します。

構文

```
public abstract String[] getParms ()
```

戻り値

固定サイズのパラメータの配列。パラメータ未使用の場合は NULL。

関連情報

ValidateDatabaseProgressData.StatusId インタフェース [258 ページ]

1.33.2 getStatusId() メソッド

検証操作のステータス ID を返します。

構文

```
public abstract short getStatusId ()
```

戻り値

ステータス ID 定数。

1.34 ValidateDatabaseProgressData.StatusId インタフェース

Ultra Light データベース検証ユーティリティのステータス ID を指定します。

構文

```
public interface StatusId
```

メンバー

StatusId のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

変数

変更子とタイプ	変数	説明
public final short	UL_VALID_NO_ERROR	エラーは発生しませんでした。
public final short	UL_VALID_START	検証を開始します。

変更子とタイプ	変数	説明
public final short	UL_VALID_END	検証を終了します。 ValidateDatabaseProgressData.getParms メソッドによって返される最初のパラメータは、成功または失敗を示す結果の SQLCODE を追跡します。
public final short	UL_VALID_CHECKING_PAGE	データベースページのチェック中、定期的にステータスメッセージを送信します。 ValidateDatabaseProgressData.getParms メソッドから返される最初のパラメータは、ページに関連付けられた番号を追跡します。順序は定義されていません。
public final short	UL_VALID_CHECKING_TABLE	テーブルをチェックしています。 ValidateDatabaseProgressData.getParms メソッドから返される最初のパラメータは、テーブル名を追跡します。
public final short	UL_VALID_CHECKING_INDEX	インデックスをチェックしています。 ValidateDatabaseProgressData.getParms メソッドから返される最初のパラメータは、テーブル名を格納します。2 番目のパラメータは、インデックス名を格納します。
public final short	UL_VALID_DATABASE_ERROR	データベースにアクセスするときにエラーが発生しました。 詳細については、SQLCODE を参照してください。
public final short	UL_VALID_STARTUP_ERROR	低レベルアクセスを目的とするデータベースの起動中にエラーが発生しました。
public final short	UL_VALID_CONNECT_ERROR	データベースへの接続中にエラーが発生しました。
public final short	UL_VALID_INTERRUPTED	検証プロセスが中断されました。
public final short	UL_VALID_CORRUPT_PAGE_TABLE	ページテーブルが破損しています。
public final short	UL_VALID_FAILED_CHECKSUM	ページチェックサムが失敗しました。 ValidateDatabaseProgressData.getParms メソッドから返される最初のパラメータは、ページに関連付けられた番号を追跡します。
public final short	UL_VALID_CORRUPT_PAGE	ページが破損しています。 ValidateDatabaseProgressData.getParms メソッドから返される最初のパラメータは、ページに関連付けられた番号を追跡します。

変更子とタイプ	変数	説明
public final short	UL_VALID_ROWCOUNT_MISMATCH	インデックス内のローの数がテーブルのロー数と異なります。 ValidateDatabaseProgressData.getParms メソッドから返される最初のパラメータは、テーブル名を追跡します。2 番目のパラメータは、インデックス名を追跡します。
public final short	UL_VALID_BAD_ROWID	インデックス内に無効なロー識別子があります。 ValidateDatabaseProgressData.getParms メソッドから返される最初のパラメータは、テーブル名を追跡します。2 番目のパラメータは、インデックス名を追跡します。

1.35 ValidateDatabaseProgressListener インタフェース

ValidateDatabase プログレスイベントを受信します。

構文

```
public interface ValidateDatabaseProgressListener
```

メンバー

ValidateDatabaseProgressListener のすべてのメンバー (継承されたメンバーも含みます) を次に示します。

メソッド

変更子とタイプ	メソッド	説明
public boolean	validateProgressed(ValidateDatabaseProgressData) [261 ページ]	ユーザに検証の進行状況を通知するために、ValidateDatabase 操作中に呼び出されます。

このセクションの内容:

validateProgressed(ValidateDatabaseProgressData) メソッド [261 ページ]
ユーザに検証の進行状況を通知するために、ValidateDatabase 操作中に呼び出されます。

1.35.1 validateProgressed(ValidateDatabaseProgressData) メソッド

ユーザに検証の進行状況を通知するために、ValidateDatabase 操作中に呼び出されます。

構文

```
public boolean validateProgressed (ValidateDatabaseProgressData data)
```

パラメータ

data 最新の検証プログレスデータを保持している ValidateDatabaseProgressData オブジェクト。

戻り値

検証プロセスをキャンセルする場合は true、そうでない場合は false。

2 このマニュアルの印刷、再生、および再配布

次の条件に従うかぎり、このマニュアルの全部または一部を使用、印刷、再生、配布することができます。

1. ここに示したものとそれ以外のすべての版権と商標の表示をすべてのコピーに含めること。
2. マニュアルに変更を加えないこと。
3. SAP 以外の人間がマニュアルの著者または情報源であるかのように示す一切の行為をしないこと。

ここに記載された情報は事前の通知なしに変更されることがあります。

重要免責事項および法的情報

コードサンプル

この文書に含まれるソフトウェアコード及び / 又はコードライン / 文字列 (「コード」) はすべてサンプルとしてのみ提供されるものであり、本稼動システム環境で使用することが目的ではありません。「コード」は、特定のコードの構文及び表現規則を分かりやすく説明及び視覚化することのみを目的としています。SAP は、この文書に記載される「コード」の正確性及び完全性の保証を行いません。更に、SAP は、「コード」の使用により発生したエラー又は損害が SAP の故意又は重大な過失が原因で発生させたものでない限り、そのエラー又は損害に対して一切責任を負いません。

アクセシビリティ

この SAP 文書に含まれる情報は、公開日現在のアクセシビリティ基準に関する SAP の最新の見解を表明するものであり、ソフトウェア製品のアクセシビリティ機能の確実な提供方法に関する拘束力のあるガイドラインとして意図されるものではありません。SAP は、この文書に関する一切の責任を明確に放棄するものです。ただし、この免責事項は、SAP の意図的な違法行為または重大な過失による場合は、適用されません。さらに、この文書により SAP の直接的または間接的な契約上の義務が発生することは一切ありません。

ジェンダーニュートラルな表現

SAP 文書では、可能な限りジェンダーニュートラルな表現を使用しています。文脈により、文書の読者は「あなた」と直接的な呼ばれ方をされたり、ジェンダーニュートラルな名詞 (例: 「販売員」又は「勤務日数」) で表現されます。ただし、男女両方を指すとき、三人称単数形の使用が避けられない又はジェンダーニュートラルな名詞が存在しない場合、SAP はその名詞又は代名詞の男性形を使用する権利を有します。これは、文書を分かりやすくするためです。

インターネットハイパーリンク

SAP 文書にはインターネットへのハイパーリンクが含まれる場合があります。これらのハイパーリンクは、関連情報を見いだすヒントを提供することが目的です。SAP は、この関連情報の可用性や正確性又はこの情報が特定の目的に役立つことの保証は行いません。SAP は、関連情報の使用により発生した損害が、SAP の重大な過失又は意図的な違法行為が原因で発生したものでない限り、その損害に対して一切責任を負いません。すべてのリンクは、透明性を目的に分類されています (http://help.sap.com/disclaimer を参照)。

go.sap.com/registration/contact.html